Debt-Aware Bonding Curves: Rising Floor Prices and Non-Liquidatable Borrowing

A Whitepaper

Ömer Demirel* Floors Finance

Michael Lewkowitz† House Markets

Tiago Santana‡ Floors Finance

March 2026

Abstract

Decentralized lending protocols rely on external price oracles and liquidation mechanisms to maintain solvency. When collateral prices drop sharply, liquidations cascade—triggering forced sales that depress prices further and inflict losses on borrowers and the broader market.

This paper introduces *debt-aware bonding curves*: a token issuance mechanism with a built-in price floor that can only go up. Tokens are minted and redeemed through a stepped bonding curve whose lowest tier—the floor—guarantees a minimum redemption price backed by onchain reserves. A reserve invariant couples the curve's collateral to outstanding debt, enabling a credit facility for *issuance-native collateral* in which borrowing capacity is anchored to the endogenous floor price rather than a market oracle. Because the floor price is monotonically non-decreasing, no loan originated within the floor-anchored LTV can become under-collateralized due to collateral price declines—eliminating protocol-triggered liquidation. The trade-off: non-repayment results in permanent token lock, not forced sale.

We describe the mechanism, explain why every loan remains safe without liquidation, and show how a recursive buy-lock-borrow-buy loop enables non-liquidatable leveraged positions—particularly useful for token launches where no external oracle exists. The safety properties are proved mathematically and verified at the implementation level through stateful fuzz testing and formal verification (Certora CVL, Halmos); economic performance is evaluated via agent-based simulation.

*omer@floors.finance

†michael@house.markets

‡tiago@floors.finance

1 Introduction

1.1 The Liquidation Problem

Decentralized lending protocols—Aave [Aave, 2020], Compound [Leshner and Hayes, 2019], and MakerDAO[1] [MakerDAO, 2017]—allow users to borrow against onchain collateral. They all share the same design: an external price oracle determines the value of the collateral, and a liquidation mechanism forcibly closes positions when that value drops below a safety threshold.

This architecture creates systemic risk. When prices fall, liquidations generate cascading sell pressure that pushes prices down further, triggering more liquidations [Perez et al., 2021, Tian and Zhu, 2025]. During the March 2020 "Black Thursday" event, over $8 million in MakerDAO positions were liquidated, with some collateral sold for near-zero bids [Perez et al., 2021]. Research from the Bank for International Settlements has documented how DeFi leverage amplifies volatility across protocols [Heimbach and Huang, 2024, Aramonte et al., 2022], and liquidation events attract MEV extraction as searchers compete to front-run liquidation transactions [Daian et al., 2020].

1.2 A Different Approach

Several protocols have explored alternatives. Nirvana [Nirvana Finance, 2022] introduced a virtual AMM with a ratcheting floor price on Solana, but was drained of $3.5M via a flash loan exploit. Baseline [Baseline Markets, 2024] enforces a reserve-backed floor through a proprietary market maker. Olympus [Olympus DAO, 2021] backs its token with a diversified treasury and uses automated monetary policy to defend a backing price. Each represents a different point in the design space; to our knowledge, none of these designs provides formal monotonicity proofs for the floor price or invariant-based safety guarantees for loans.

Our mechanism takes a different path. Instead of relying on external oracles, we use a *bonding curve* as a primary-market token issuer: tokens are minted on purchase and burned on sale, with collateral flowing into and out of a protocol-controlled reserve. The design builds on the discrete bonding curve (DBC) [Demirel, 2023a] and the dynamic discrete bonding curve (DDBC) [Demirel, 2023b], which established stepped bonding curves as gas-efficient, protocol-owned issuance mechanisms [Demirel, 2024]. We extend these foundations with a *debt-aware* reserve invariant and a *floor segment* whose price is provably monotonic, enabling non-liquidatable borrowing as a direct consequence of the curve's mathematical properties.

Because the protocol controls the entire token supply through mint/burn, it can enforce a contractual *floor price*—a guaranteed minimum redemption price backed by onchain reserves. We term this *token-owned liquidity*: the reserve backing each token is embedded in its issuance mechanism rather than held in an external treasury or rented from liquidity providers. A key economic consequence is that trading fees and origination fees—value streams that would ordinarily accrue to external market makers, AMM liquidity providers, or lending protocols—are captured by the token's own issuance contract and directed toward floor elevation, creating a self-reinforcing value accrual loop.

[1] MakerDAO rebranded to Sky (`https://sky.money`) in 2024.

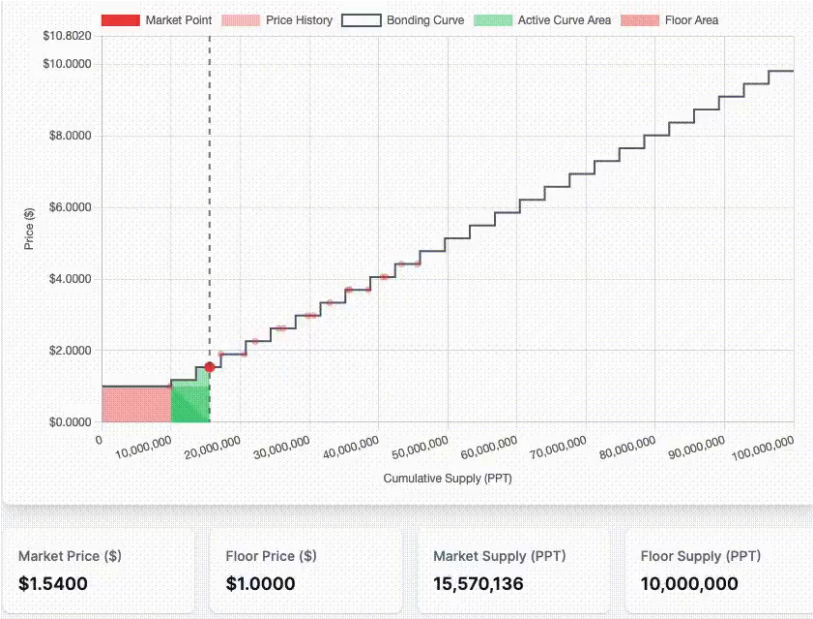

Figure 1: Overview of a debt-aware bonding curve. The *floor area* (pink) represents the virtual collateral backing unlocked tokens at the floor price P_f (when debt is outstanding, actual reserves cover free supply; see Section 3). The *active curve area* (green) shows the premium segments where new tokens are minted at progressively higher prices. The market point (red dot) marks the current supply and spot price.

Scope. The guarantees described in this paper apply to *issuance-native collateral*: tokens minted through the protocol's own bonding curve and redeemable at its primary-market floor price. The mechanism does not replace general-purpose lending against arbitrary external assets. Secondary markets may temporarily trade below the floor; convergence depends on arbitrageur capital and transaction costs (Section 6).

1.3 What This Paper Covers

This whitepaper explains how debt-aware bonding curves work and why they eliminate the need for liquidation. Specifically:

- How the stepped bonding curve and floor price mechanism work (Sections 2–3).
- Why the floor price can only go up and every loan stays safe without liquidation (Section 4).
- How a recursive leverage loop enables non-liquidatable leveraged positions, with token launches as the primary application (Section 4.5).
- How the mechanism has been verified through fuzz testing, formal verification, and simulation (Section 5).
- Known limitations and trade-offs (Section 6).

For full formal proofs and mathematical details, see the companion academic paper [Ömer Demirel et al., 2026]. Verification artifacts (Certora CVL specifications, Halmos test contracts) are available at `https://github.com/demirelo/dabc-proofs`.

2 How It Works

This section provides a visual overview of the debt-aware bonding curve before the detailed mechanism description in Section 3.

2.1 The Stepped Bonding Curve

A debt-aware bonding curve is a price function composed of discrete *segments*. The first segment—the *floor segment*—is flat and establishes a guaranteed minimum redemption price P_f. Subsequent *premium segments* step upward in price, forming a price ladder for token issuance.

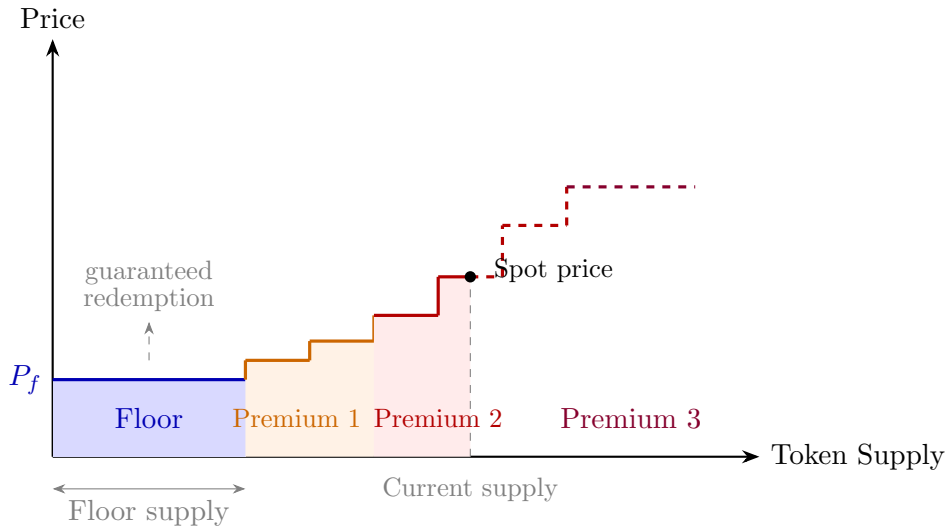

Figure 2: A debt-aware bonding curve with four segments. The floor segment (blue) guarantees a minimum redemption price P_f. Premium segments form a stepped price ladder. Solid fills and lines show minted supply; dashed lines show unminted capacity available for future purchases. Every *unlocked* token can be redeemed through the curve at its step price or at P_f if supply retracts to the floor. Locked tokens (loan collateral) are held in custody until the associated loan is repaid.

Every token is minted *through* the curve and, while unlocked, can be redeemed back through it. Because the protocol controls the entire supply via mint/burn, the floor price is contractually enforceable on the primary market.

2.2 Protocol Operations

The mechanism supports five core operations, shown in Figure 3.

- **Buy** mints new tokens and deposits collateral into the reserve.
- **Sell** burns tokens and withdraws collateral.
- **Lock** transfers tokens into the credit facility's custody.
- **Borrow** withdraws reserve collateral against locked tokens, with borrowing power anchored to the floor price.
- **Repay** returns collateral and unlocks tokens.

"Non-liquidatable" means: if a borrower does not repay, the tokens remain locked indefinitely. The consequence of non-repayment is *permanent lock, not forced sale*. No protocol operation forcibly seizes or sells a borrower's position.

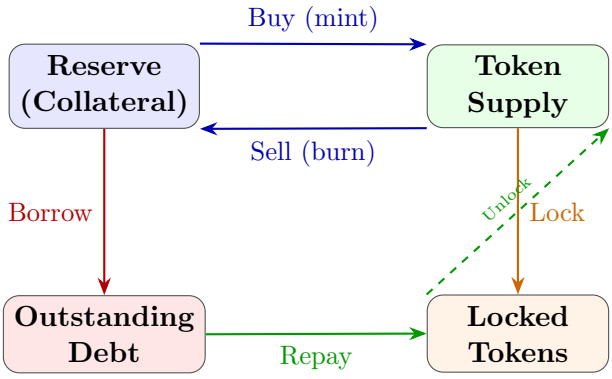

Figure 3: Protocol operations and state transitions. Buy deposits collateral and mints tokens. Sell burns tokens and withdraws collateral. Lock places tokens in credit facility custody. Borrow withdraws collateral at a fraction of the floor-price value. Repay returns collateral and unlocks tokens.

2.3 Floor Elevation

The floor price P_f can be raised by injecting collateral—for example, from trading fees or surplus revenue—into the floor segment. The *step-absorption* process raises P_f by merging premium steps into the floor, as shown in Figure 4.

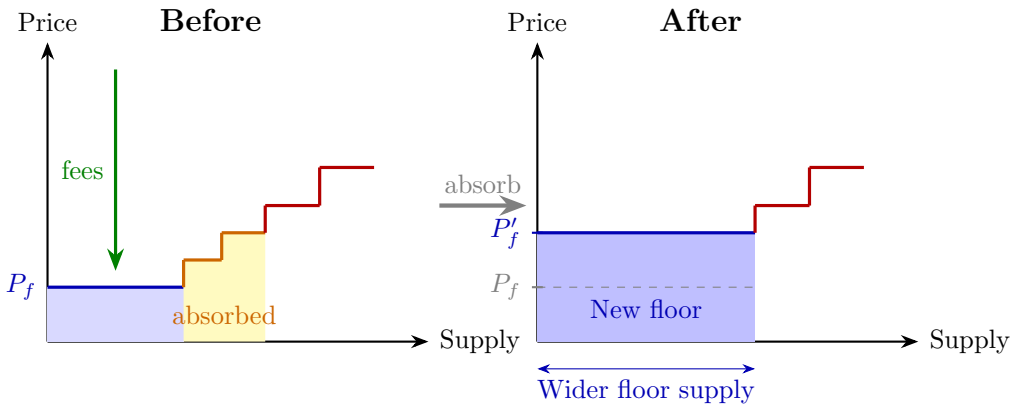

Figure 4: Step absorption raises the floor price. **Left:** fees are injected into the floor reserve. The yellow-shaded premium steps are absorbed. **Right:** absorbed steps merge into the floor, yielding a higher floor price $P'_f \geq P_f$ and a wider floor supply. Remaining premium segments are unchanged.

The floor price is *monotonically non-decreasing*: no protocol operation can lower P_f. This is the property that makes borrowing non-liquidatable—the value anchor for borrowing power can only stay the same or improve.

2.4 The Leverage Loop

The credit facility enables a recursive leverage strategy: buy tokens, lock them, borrow collateral at a fraction of their floor value, and reinvest the proceeds. Each iteration reinvests a fraction of the previous round's capital, and the total leverage converges to a bounded maximum (Figure 5).

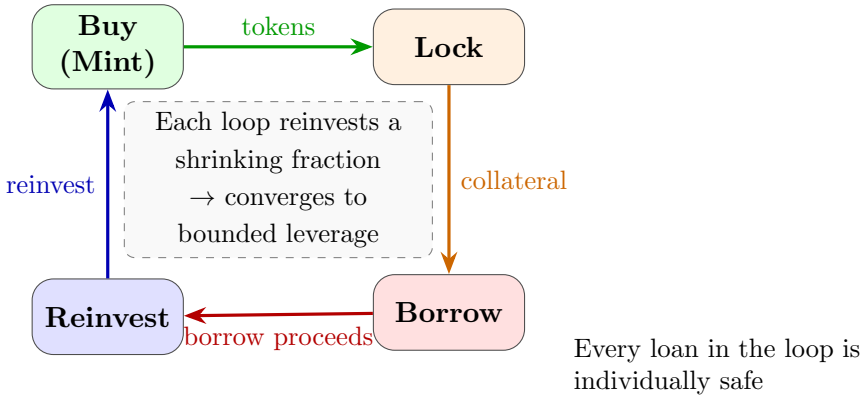

Figure 5: The recursive leverage loop. Each iteration reinvests a shrinking fraction of the previous round's capital. Because borrowing is anchored to the floor price, every loan in the loop is safe—no iteration can produce an under-collateralized position.

Token launches are the most compelling application: the floor-to-spot gap is smallest early in the curve, maximizing looping efficiency, and no external oracle exists for newly issued tokens—making the floor-anchored credit facility the only available source of non-liquidatable leverage.

3 Mechanism Design

This section explains the core mechanics of the debt-aware bonding curve: how the price ladder is structured, how the reserve ensures solvency, how the floor is raised, and how the curve can be reshaped.

3.1 The Stepped Price Curve

The bonding curve is a sequence of *segments*, each defining a price tier. A segment has four parameters:

- **Initial price**: the starting price of the tier.
- **Price increment**: how much the price increases per step within the tier (zero for flat tiers).
- **Supply per step**: how many tokens are sold at each step price.
- **Number of steps**: how many steps the tier contains.

The *floor segment* is always the first segment. It is flat (price increment = 0) with exactly one step, establishing the guaranteed minimum redemption price P_f. Premium segments follow, forming ascending price tiers.

Unlike secondary-market AMMs (e.g., Uniswap) that trade pre-existing token inventories, this bonding curve is a *primary-market* issuer. Buying mints new tokens and deposits collateral; selling burns tokens and withdraws collateral. Because the protocol controls the entire supply, the floor price is enforceable: every token minted through the curve can be redeemed through it.

3.2 The Reserve and Solvency

The *reserve function* computes the total virtual collateral needed to back all outstanding tokens at their respective step prices. It works by walking through the segments in order, multiplying price × quantity at each step, and summing the results.

The Solvency Rule

At every state transition, the protocol's virtual collateral supply V equals the reserve computed from the current segment configuration and token supply:

$$V \;=\; R(\Gamma,\, S)$$

Every buy adds collateral, every sell returns it, and the accounting is checked atomically. The protocol never under-collateralizes.

The *virtual* collateral supply V is an onchain counter that tracks total backing. During borrowing, the contract's *actual* balance may be lower ($A = V - D$, where D is total outstanding debt), because some collateral has been lent out. The locked tokens backing those loans guarantee the reserve can be restored upon repayment.

Locked vs. free supply. When tokens are locked as loan collateral, they are transferred to the credit facility's custody and cannot be sold or redeemed. The remaining *free* (unlocked) supply is $S_{\text{free}} = S - \sum_i \ell_i$. The actual reserves $A = V - D$ only need to cover redemptions of this free supply—not the locked portion. This works because each loan satisfies $d_i < P_f \cdot \ell_i$ (the locked tokens are worth more at floor than the debt they back), so repayment fully restores the virtual reserves.

Worked example. Consider a curve with three segments:

- **Floor**: price 100, capacity 1,000 tokens.
- **Premium 1**: step prices 120, 130, 140 (three steps of 500 tokens each); capacity 1,500 tokens.
- **Premium 2**: price 200, capacity 2,000 tokens.

If 2,200 tokens are outstanding, the floor absorbs 1,000 tokens (reserve: $1{,}000 \times 100 = 100{,}000$). The remaining 1,200 fall in Premium 1: two full steps at prices 120 and 130 (reserve: $500 \times 120 + 500 \times 130 = 125{,}000$), plus 200 tokens at the third step price of 140 (reserve: $200 \times 140 = 28{,}000$).

Total reserve: $100{,}000 + 125{,}000 + 28{,}000 = 253{,}000$. The virtual collateral supply must equal 253,000 for solvency to hold.

Implementation Note

All formulas are stated in abstract units where price × quantity = collateral directly. The Solidity implementation uses 18-decimal fixed-point arithmetic (WAD), with ceiling rounding on collateral computations to ensure the contract never under-collateralizes at wei-level precision.

3.3 Raising the Floor

When external collateral is injected—typically from trading fees or surplus revenue—the floor price can be raised through *step absorption*. The process works as follows:

1. Calculate how much the injected collateral can raise the floor price, given the current floor supply.
2. If the raise reaches the next premium step, absorb that step: the floor price jumps to the premium step's price, and the floor supply grows by the step's capacity.
3. Repeat until the collateral budget is exhausted or all premium steps are absorbed.
4. Recompute the exact reserve to ensure solvency, and transfer only the precise amount needed.

The process operates in two modes. In *normal mode* (floor supply < actual token supply), absorbing a step increases both the floor price and supply. In *capped mode* (floor supply = actual supply), only the price increases; the protocol does not back unminted tokens.

Floor Price Monotonicity

No protocol operation—buy, sell, borrow, repay, raise floor, or reconfigure—can lower the floor price. Once the floor goes up, it stays up. This is the foundational guarantee that makes non-liquidatable borrowing possible.

Harmonic step sizing. The cost of raising the floor by one price unit is proportional to the floor supply. If all premium steps have equal capacity, the floor supply grows linearly and elevation costs scale quadratically. Using decreasing step capacities (e.g., halving with each tier) keeps the floor supply growth logarithmic and elevation costs manageable over time.

3.4 Curve Reconfiguration

The premium segments above the floor can be reshaped by governance, subject to three constraints:

1. **Reserve invariance**: the new curve must require the same total collateral for the current supply. Any shortfall must be supplied alongside the reconfiguration.
2. **Bounded spot-price loss**: no minted token's price may decrease by more than a configured maximum $\delta_{\max}$ per invocation. Setting $\delta_{\max} = 0$ makes reconfiguration fully non-dilutive.
3. **Cumulative dilution budget**: an onchain counter tracks total spot-price loss across all reconfigurations. A reconfiguration is rejected if it would cause the cumulative loss to exceed $B_{\max}$.

This makes the bonding curve a *configurable liquidity primitive*: the shape of premium tiers can adapt to market conditions while the floor guarantee and reserve solvency are preserved. We call authorized reconfigurations *Liquidity Reallocation Events (LREs)*.

Floor elevation is a special case of reconfiguration that satisfies all constraints by construction—it only raises prices.

Governance risk warning. Reconfiguration is the largest governance-controlled attack surface. Token holders should be aware that each invocation may reduce spot prices by up to $\delta_{\max}$, and $B_{\max}$ caps the total dilution. The reserve invariant protects solvency and the cumulative budget protects against unbounded dilution, but neither alone prevents all forms of value redistribution. Deployments should pair reconfiguration authority with timelocks, multisig requirements, and rate limits.

3.5 Why Discrete Steps?

The choice of discrete (stepped) pricing over continuous curves (e.g., polynomial or exponential) is deliberate.

Table 1: Discrete vs. continuous bonding curves.

Property	**Discrete**	**Continuous**
Intra-step slippage	None	Proportional to order size
Floor segment	Native (flat step)	Requires auxiliary contract
Onchain math	Integer arithmetic	Transcendental functions
Floor elevation	Step iteration	Integral solving
Rounding control	Per-step boundaries	Distributed error
Formal verification	Finite state space	Infinite state space
Granularity	Arbitrarily small steps	Inherently smooth

The key advantages are deterministic pricing (no slippage within a step), native floor support, integer-only arithmetic (no transcendental functions), and amenability to formal verification. By choosing sufficiently small step sizes, the discrete curve approximates any continuous price function while retaining these benefits.

4 Non-Liquidatable Borrowing

4.1 How Borrowing Works

Token holders can borrow the reserve asset (e.g., USDC, ETH) against their holdings. The process is straightforward:

1. **Lock** tokens into the credit facility's custody.
2. **Borrow** up to $\gamma \times P_f \times \ell$ collateral, where γ is the loan-to-value ratio (e.g., 0.80), P_f is the current floor price, and ℓ is the number of locked tokens.
3. **Repay** the borrowed amount to unlock your tokens.

A one-time origination fee is charged at loan creation. There is no streaming interest—debt does not grow over time.

What happens if you don't repay? The tokens remain locked indefinitely. They are not liquidated, not auctioned, and not forcibly sold. The consequence of non-repayment is permanent lock, not forced sale. This is a deliberate design choice: any mechanism that forcibly closes positions reintroduces liquidation risk.

Why no interest? This is not a simplification—it is a prerequisite. If interest accrued continuously at rate r, the debt would grow as $d(t) = d_0\, e^{r(t-t_0)}$, eventually requiring P_f to rise at the same rate to maintain safety. Since floor elevation depends on fee revenue that cannot be guaranteed, no protocol can ensure this under all conditions. By fixing debt at origination, the loan can only get safer over time as P_f rises.

Non-rivalrous borrowing power. Unlike Aave or Compound, where all borrowers draw from a shared pool and one borrower's utilization reduces the credit limit available to others, each borrower's maximum borrowing power is derived solely from their own locked tokens. One borrower's loan does not reduce another's credit limit. However, borrow *execution* depends on the contract's current collateral balance: if outstanding loans have already withdrawn a large fraction of the physical reserve, a new borrow may revert until liquidity is restored through repayments. This is a transient availability constraint, not an insolvency event—the virtual collateral still covers all obligations.

4.2 End-to-End Lifecycle Example

We trace a single position through the full protocol lifecycle.

Setup. $P_f = \$1.00$, $S_0 = 5{,}000$, $\gamma = 0.80$, $\phi = 0.02$ (origination fee). Alice has \$500 in reserve tokens.

Step	Action and state change
1	**Mint.** Alice buys 500 tokens at \$1.00 each, depositing \$500 into the reserve. Supply increases to 5,500; virtual collateral is updated.
2	**Lock & borrow.** Alice locks all 500 tokens and borrows $0.80 \times 1.00 \times 500 = \$400$, minus origination fee $0.02 \times 400 = \$8$, receiving \$392. Loan: ($\ell = 500$, $d = 400$). Actual reserves decrease by \$400, but virtual reserves are unchanged. The locked tokens back the debt: $500 \times \$1.00 = \$500 > \$400$.
3	**Floor raise.** Protocol fees accumulate and step absorption raises P_f to \$1.20. Alice's debt is still \$400, but her collateral is now valued at $500 \times \$1.20 = \$600$. Effective LTV drops from 80% to $400/600 = 66.7\%$ (passive de-risking).
4	**Rebalance (optional).** Alice's new maxBorrow $= 0.80 \times 1.20 \times 500 = \$480$. She can borrow an additional $\$480 - \$400 = \$80$ (minus fees) without adding collateral.
5	**Repay.** Alice repays \$400. The protocol unlocks all 500 tokens proportionally. Loan closed; actual reserves restored; tokens freely tradable.
6	**Redeem.** Alice sells 500 tokens at market price ($\geq P_f = \$1.20$), receiving $\geq \$600$. Supply returns to 5,000.

Key observations. (i) At no point does the protocol require a price oracle; all valuations use the endogenous P_f. (ii) The floor raise at step 3 improves Alice's position without any action on her part. (iii) If Alice *never* repays, her 500 tokens remain locked permanently—the permanent-lock trade-off—but solvency and floor monotonicity are unaffected.

4.3 Why Every Loan Stays Safe

The safety argument rests on two properties:

Two Guarantees

1. **Floor price only goes up.** No protocol operation can lower P_f. Buy, sell, borrow, repay, floor raise, and reconfiguration all preserve or increase the floor.
2. **Debt is always less than the floor value of locked tokens.** At origination, $d \leq \gamma \cdot P_f \cdot \ell$ with $\gamma < 1$, so $d < P_f \cdot \ell$. Since d is fixed (no interest) and P_f can only increase, the gap between debt and collateral value only widens over time.

Passive de-risking. As the floor price rises through fee accumulation and step absorption, every existing loan becomes better collateralized without any action from the borrower. The effective loan-to-value ratio decreases over time:

$$\text{Effective LTV} \;=\; \frac{\text{remaining debt}}{P_f(\text{now}) \times \ell} \;\leq\; \gamma \cdot \frac{P_f(\text{origination})}{P_f(\text{now})} \;\leq\; \gamma$$

This is the opposite of oracle-based lending, where capital efficiency degrades with volatility.

LTV changes don't affect existing loans. Even if governance lowers γ after a loan is created, the safety bound $d < P_f \cdot \ell$ depends only on $\gamma < 1$ at origination. Existing loans remain safe regardless of future parameter changes.

Sell-below-floor and outstanding loans. If redemptions reduce supply below the floor supply, the protocol adjusts the floor supply to match while keeping P_f fixed. Locked tokens from outstanding loans cannot be sold (they are held in custody), so actual reserves only need to cover redemptions of *unlocked* supply. All loan safety invariants are preserved through this adjustment.

4.4 Comparison with Existing Protocols

Table 2 contrasts the proposed credit facility with established DeFi lending protocols.

Table 2: Comparison of borrowing designs.

Property	**This mechanism**	**Aave / Compound**	**MakerDAO**
Collateral valuation	Floor price P_f	Spot oracle	Spot oracle
Price monotonicity	Guaranteed	Not guaranteed	Not guaranteed
Liquidation	None	Health-factor auction	Keeper auction
Interest	None (one-time fee)	Variable / stable rate	Stability fee
Loan term	Open-ended	Open-ended	Open-ended
Oracle dependency	None (endogenous)	External price feeds	External price feeds

4.5 Recursive Leverage

The credit facility enables a recursive strategy: buy tokens, lock them, borrow against the floor, and use the proceeds to buy more tokens. This loop can be repeated to amplify exposure.

Let γ be the LTV ratio and ϕ the effective fee per iteration. Each loop reinvests a fraction $\eta = \gamma \cdot (1 - \phi)$ of the previous round, so the total leverage converges to:

$$L_{\text{net}} \;=\; \frac{1}{1-\eta} \;=\; \frac{1}{1-\gamma(1-\phi)}$$

Worked example: step-by-step leverage loop. Suppose $P_f = \$1.00$, buying at the floor, $\gamma = 0.90$, $\phi = 0.03$, and initial capital $C_0 = \$1{,}000$.

Loop	Action	Capital in	Tokens	Borrow	Debt
0	Buy	\$1,000	1,000	—	—
0	Lock + Borrow	—	—	\$873	\$873
1	Buy	\$873	873	—	—
1	Lock + Borrow	—	—	\$762	\$762
2	Buy	\$762	762	—	—
2	Lock + Borrow	—	—	\$665	\$665
3	Buy	\$665	665	—	—
	⋮	⋮	⋮	⋮	⋮
∞	Total	\$7,874	7,874	—	\$6,874

At each iteration, the borrower receives $\gamma(1-\phi) = 0.873$ of the previous round's capital. The total converges to approximately \$7,874 in tokens, with cumulative debt $\approx$ \$6,874. Every individual loan satisfies $d_i < P_f \cdot \ell_i$ (since $\gamma = 0.90 < 1$), so no loan in the loop can become under-collateralized.

Table 3: Net leverage for selected parameter combinations.

γ (LTV)	ϕ (fee)	η	Leverage
0.90	3%	0.873	7.87×
0.90	4%	0.864	7.35×
0.85	3%	0.825	5.70×
0.80	3%	0.776	4.46×
0.95	3%	0.922	12.74×

Natural leverage decay. On the discrete curve, each successive buy encounters higher-priced premium steps. The effective reinvestment fraction per iteration decays as $\eta_i = \gamma(1-\phi) \cdot P_f / P_{\text{spot}}^{(i)}$, where $P_{\text{spot}}^{(i)} \geq P_f$ increases with each purchase. Realized leverage is therefore always strictly below the theoretical maximum L_{net}. This is a strength: leveraged supply expansion is algorithmically self-limiting without external circuit breakers.

Safe Leverage

Every loan created in the leverage loop is individually safe. The solvency check is verified at every iteration—if remaining collateral is insufficient, the loop terminates. The loop cannot create bad debt.

4.6 Application: Leveraged Token Launches

Token launches are the most compelling use case for recursive leverage:

- The floor-to-spot gap is smallest early in the curve, maximizing looping efficiency.
- No external oracle exists for newly issued tokens, making the floor-anchored credit facility the only available source of non-liquidatable leverage.
- Each loop iteration generates origination fees that are directed to the floor reserve, funding floor elevation. This replaces inflationary token emissions commonly used to

bootstrap DeFi protocols.

Fee-funded bootstrapping. The one-time origination fee serves a dual role: beyond pricing credit risk, it functions as an algorithmic bootstrapping mechanism. Each leverage loop iteration generates fees directed to the floor reserve, funding floor elevation. Rather than diluting existing holders to attract liquidity (the standard approach in DeFi), the mechanism channels leverage demand into floor appreciation that benefits all token holders—a non-dilutive bootstrapping loop.

Anti-front-running. During presale periods, a time-decaying fee multiplier is applied. Fees start high (e.g., 5× the base rate) and decay to 1× over the presale duration, following a quartic curve that sheds over 90% of the premium by the midpoint. This discourages front-running by ensuring the highest fees are encountered immediately upon presale opening.

5 Verification and Evidence

The mechanism's safety properties are supported by multiple independent verification methods applied to a concrete Solidity implementation (smart contracts and testing framework licensed under BUSL-1.1).

Evidence Hierarchy

- **Mathematically proved** (companion paper [Ömer Demirel et al., 2026]): floor monotonicity, non-liquidatable safety, redemption liquidity, reconfiguration safety, and leverage bounds. These hold for any correct implementation under the stated assumptions.
- **Formally verified** (Certora CVL + Halmos): 24 verification rules across 6 property groups via Certora CVL (algebraic reasoning); 8 properties independently confirmed via Halmos symbolic execution.
- **Fuzz-tested** (Foundry stateful invariant tests): reserve solvency, floor monotonicity, segment validity, collateral backing, and loan safety across randomized operation sequences.
- **Simulated** (agent-based model): dynamic fee performance across market regimes, floor growth trajectories, leverage convergence.
- **Indicative only**: gas cost measurements (local testnet, compiler-version-dependent).

Three distinct axes. The mechanism's properties decompose into three axes that should not be conflated:

- **Safety**: the reserve invariant holds and no loan becomes under-collateralized. Safety holds regardless of repayment behavior or activity level.
- **Liquidity**: actual reserves $A = V - D$ cover all free-supply redemptions. Liquidity is temporarily reduced when debt is outstanding—the reserve is solvent but partially deployed.
- **Efficiency**: borrowing capacity per locked token grows as P_f rises. Efficiency improves passively but depends on fee revenue; dormant loans drag on floor-elevation velocity

Figure 6: Median simulation run: floor price (blue) and market price (pink) over approximately one year. The floor price rises monotonically as trading fees are absorbed, while the market price fluctuates freely above the floor.

without affecting safety.

Safe does not imply liquid in the pooled sense; liquid does not imply efficient.

5.1 Invariant-Based Fuzz Testing

The implementation is verified through stateful fuzz testing using the Foundry framework. Handler contracts wrap each protocol module and expose randomized sequences of protocol operations. After every operation sequence, the following invariants are checked:

1. **Reserve solvency**: the virtual collateral supply matches the reserve computed from the current segments and supply.
2. **Floor monotonicity**: the floor price never falls below its deployment value.
3. **Segment validity**: the curve always has at least two segments; the floor segment has one step with zero price increment.
4. **Collateral backing**: the actual token balance plus cumulative withdrawals equals the virtual supply plus cumulative deposits.
5. **Loan safety**: every active loan satisfies $d \leq \gamma \cdot P_f \cdot \ell$, and the sum of locked tokens matches the credit facility's balance.

Beyond per-module invariant tests, a full-protocol workflow handler orchestrates multi-step scenarios—buy-and-borrow, full loan lifecycles (borrow $\rightarrow$ rebalance $\rightarrow$ repay), leveraged loop positions, and concurrent multi-user sequences.

5.2 Formal Verification

The core contract has been formally verified using two independent tools:

- **Certora CVL**: 24 verification rules across 6 property groups—segment validity (5 rules), floor monotonicity (3), reserve solvency (5), loan safety (3), repayment ratio preservation (3), and buy/sell safety (5). The Certora Prover uses algebraic reasoning, making it well-suited for the division-heavy and `mulDiv`-based properties central to

reserve solvency and loan safety.

- **Halmos**: 8 symbolically executed properties via bounded model checking (Z3 SMT solver), independently confirming segment validity, floor monotonicity, reserve solvency, loan safety, and repayment safety. The remaining 16 properties involve 256-bit integer division that exceeds Z3's bitvector timeout; these are covered by the Certora specifications.

All specifications are publicly available at https://github.com/demirelo/dabc-proofs.

5.3 Comparative Analysis

Table 4 positions the mechanism against representative floor-price and reserve-backed token designs.

Table 4: Comparison of floor-price and reserve-backed designs. • = present, ○ = absent, △ = partial.

Property	**This work**	**Nirvana**	**Baseline**	**Olympus**	**Aave**
Floor enforcement	•	•	•	○	○
Guarantee type	Reserve inv.	Algorithmic	Algorithmic	Treasury	N/A
Borrowing	•	•	•	○	•
Oracle dependency	○	○	○	•	•
Floor monotonicity	Proven	Claimed	Claimed	○	N/A
Reconfigurability	•	○	○	○	N/A

The key differentiators are: (1) the floor guarantee is backed by a reserve invariant with formal proofs, not just algorithmic claims; (2) the curve can be reconfigured while preserving all safety properties; and (3) no external oracle is required.

5.4 Adversarial Stress Analysis

Three adversarial scenarios target the core invariants:

Table 5: Adversarial scenario summary.

Scenario	**Attack path**	**Defence**	**Residual risk**
Reconfig. exploitation	Repeated $\delta_{\max}$ reconfigs	Cumulative budget $B_{\max}$; floor only rises	Spot loss $\leq B_{\max}$ for premium holders
Reserve drain	All holders lock + borrow at γ	Actual reserves cover free supply	Zero free supply; redemptions blocked until repayment
Secondary divergence	Market price $< P_f$	Unconditional primary redemption	Slow convergence if arb. capital is low

In all three scenarios, safety properties (floor monotonicity, loan safety, reserve solvency)

are preserved. The efficiency and liquidity axes degrade gracefully under stress without compromising the core invariants.

5.5 Failure Modes

We distinguish properties that *cannot* be violated under a correct implementation from those vulnerable to implementation errors:

- **Structurally safe**: floor monotonicity, reserve solvency, and loan safety hold by construction. No sequence of protocol operations can violate them.
- **Vulnerable to implementation errors**: smart contract bugs could violate any invariant. Non-standard collateral tokens (fee-on-transfer, rebasing) would break the accounting identity. Operator misconfiguration can reshape premium segments within the $\delta_{\max}$ bound, but onchain preconditions prevent violations of the floor or reserve invariant.

5.6 Dynamic Fee Simulation

An agent-based simulation evaluated seven fee strategies across six market regimes (low/high volatility, bull/bear trend, mean-reverting, liquidity shock), with 20 independent seeds per regime (840 total simulations).

The *Floors Dynamic* fee model—combining a premium-aware multiplier with quadratic size scaling—ranked first in all six regimes, achieving a +17.1% revenue uplift over static fees. The model uses only integer arithmetic and fits in a single EVM storage slot (8 parameters, 256 bits total).

Table 6: Mean protocol revenue across 20 seeds per regime. Bold indicates the highest revenue.

Regime	**Static**	**Best alternative**	**Floors Dynamic**
Low volatility	116.8	122.0	**137.3**
High volatility	261.3	275.0	**309.1**
Bull trend	608.9	673.0	**711.5**
Bear trend	24.9	30.0	**38.2**
Mean-reverting	102.7	104.3	**107.4**
Liquidity shock	224.0	256.5	**263.6**
Grand avg.	223.1	238.9	**261.2**

5.7 Gas Costs

Table 7 reports indicative gas costs for core operations, measured on a Foundry local testnet. Single-step operations are independent of segment count. Cross-segment operations scale linearly. For practical deployments, 10–50 segments provides sufficient curve expressiveness while keeping gas comparable to standard AMM swaps. The dynamic fee multiplier adds approximately 8,000 gas per trade.

Table 7: Indicative gas costs (thousands) for core operations.

Operation	**20 segs**	**50 segs**	**100 segs**
Buy (single step)	∼85k	∼85k	∼85k
Buy (cross-segment)	∼110k	∼140k	∼190k
Sell (single step)	∼80k	∼80k	∼80k
Sell (cross-segment)	∼105k	∼135k	∼185k
raiseFloor (1 step)	∼95k	∼95k	∼95k
raiseFloor (full traverse)	∼150k	∼250k	∼420k
Reconfigure	∼120k	∼200k	∼350k
Borrow	∼90k	∼90k	∼90k
Repay	∼85k	∼85k	∼85k

6 Limitations and Trade-offs

6.1 Primary-Market-Only Guarantee

The floor price guarantee applies to the bonding curve's primary market—where tokens are minted and redeemed. **Secondary markets (decentralized exchanges, order books) may temporarily trade below P_f.** The protocol does not and cannot control prices on external venues. Rational arbitrageurs can buy on the secondary market and redeem on the primary market to capture the spread, which exerts upward pressure toward P_f. The speed of convergence depends on arbitrageur capital, transaction costs, and cross-venue latency. During extreme events (gas spikes, bridge delays, exchange downtime), the secondary price may diverge from P_f for extended periods. Readers should not interpret "floor price" as a guarantee of the token's trading price on any venue; it is a guarantee of the minimum redemption price available through the protocol's primary market.

6.2 Fee-Dependent Floor Growth

The floor rises only when trading and borrowing activity generates fee revenue that is directed to the floor reserve. In a sustained zero-activity period, P_f remains constant—it never decreases, but it does not grow either. This can be mitigated by denominating the reserve in yield-bearing assets (e.g., staked ETH, tokenized treasuries), providing baseline floor appreciation independent of trading volume. Such assets must preserve instant redeemability (e.g., liquid staking tokens rather than natively staked assets with withdrawal delays) to maintain the redemption liquidity guarantee.

6.3 Value Internalization

A structural consequence of coupling issuance, trading, and borrowing in a single contract is that the protocol captures value streams that would otherwise flow to external intermediaries. In a conventional token economy, trading fees accrue to AMM liquidity providers; origination and interest fees accrue to lending protocols; and arbitrage profits accrue to external searchers. By contrast, a bonding curve that acts as both primary issuer and credit facility internalizes both revenue channels: (i) buy and sell fees on the primary curve and

(ii) origination fees from the credit facility. These fees are directed to the protocol's reserve, funding floor elevation via step absorption. The result is a self-reinforcing loop: protocol usage generates fees, fees raise the floor, a higher floor increases collateral quality and borrowing capacity, and increased borrowing generates further fees.

6.4 Capital Efficiency Trade-off

Borrowing power is anchored to the floor price, not the spot price. When the spot price significantly exceeds the floor, the borrowable amount per token is lower than under oracle-based lending. This conservatism is intentional: it is the structural source of the non-liquidatability guarantee.

However, capital efficiency improves passively over time as the floor rises. The effective LTV decreases without borrower action, and borrowers can rebalance—extracting additional collateral from the same locked position—as the floor appreciates. Even conservative floor-anchored borrowing can be competitive in practice: typical altcoin lending on Aave and Compound operates at LTV ranges of 50–75% with liquidation risk, while this mechanism provides LTVs in a comparable range without liquidation, and the effective LTV improves monotonically over time.

6.5 Open-Ended Debt

The one-time origination fee with no streaming interest means borrowers face no time penalty for leaving debt open. This raises two concerns:

- **Weak repayment incentives.** Mitigated through utilization-aware origination fees: the fee increases as a borrower's aggregate debt approaches the LTV limit, creating a convex penalty that discourages over-concentration. Additional non-liquidation-compatible incentives include fee rebates for borrowers with active repayment histories.
- **Dormant loans.** If borrowers abandon positions (e.g., lost keys), their locked tokens inflate the floor supply, making future floor raises more expensive. A governance-gated buyout mechanism—allowing anyone to repay a dormant loan's debt and burn the freed tokens after a long inactivity period—could address this without reintroducing oracle risk. At 50% dormancy with full buyout, the floor supply halves and floor-elevation velocity doubles.

6.6 MEV at Step Boundaries

Discrete steps create deterministic price transitions at known supply thresholds, which can attract MEV attempts. We distinguish three surfaces:

- **Primary-curve MEV**: sandwich opportunities at step boundaries. A sufficient condition to make round-trip attacks unprofitable is to ensure the fee friction exceeds the step delta: $\Delta p < 2\, f\, p_j$ where f is the one-way fee rate.
- **Secondary-pool MEV**: standard AMM MEV (sandwiching, LVR) on any external Uniswap pool. This is a property of the secondary pool, not the bonding curve.
- **Cross-market coupling**: arbitrage between primary and secondary venues creates cross-market vectors. This coupling can increase protocol throughput and fee generation, but precise characterization of cross-market MEV leakage remains an open research direction.

6.7 Idle Premium Harvesting

When tokens are locked as loan collateral, the premium capital backing those tokens above the floor becomes idle—it cannot be redeemed yet contributes to the reserve. The protocol can harvest this idle premium through an invariant-preserving reconfiguration: the premium steps backing locked supply are collapsed to the floor level, freeing surplus virtual collateral that is then injected into the floor segment via step absorption, raising P_f without any external capital injection. This optimization is subject to the same safeguards as any LRE: the cumulative dilution budget $B_{\max}$ bounds total spot-price reduction, and the reserve invariance check ensures solvency.

6.8 Governance Controls

Curve reconfiguration is the largest governance-controlled attack surface in the mechanism. Production deployments should apply:

- **Default** $\delta_{\max} = 0$: restricts reconfiguration to unminted supply regions, preventing any spot-price reduction for existing holders.
- **Epoch-based rate limits**: at most one reconfiguration per epoch (e.g., 24-hour window).
- **Timelock**: all reconfigurations with $\delta_{\max} > 0$ subject to a 48–72 hour delay.
- **Cumulative budget**: an onchain counter $B_{\max}$ caps total dilution across all invocations.

The reserve invariant protects solvency and the cumulative budget protects against unbounded dilution, but operational controls are essential to bound value redistribution across holder classes.

6.9 Trust Assumptions

The mechanism assumes: (i) the smart contract implementation is correct, (ii) the collateral token conforms to the ERC-20 standard (no fee-on-transfer or rebasing behavior), and (iii) the underlying blockchain provides liveness and finality.

6.10 What This Mechanism Does Not Solve

We separate intrinsic mechanism limits from implementation limits.

Mechanism limits (fundamental to the design):

1. **General external-asset lending.** The guarantee applies only to issuance-native collateral. Lending against external assets would require oracle-based valuation.
2. **Secondary-market price support.** The floor is enforceable only through primary-market redemption.
3. **Guaranteed floor appreciation.** P_f is non-decreasing by construction but may stay flat indefinitely without fee revenue.
4. **Open-ended repayment incentives.** No interest accrues—an essential assumption for non-liquidatable safety, but one that weakens repayment urgency.

Implementation limits (addressable in engineering scope):

1. **Smart contract risk.** All guarantees are conditional on a correct implementation.

2. **Collateral token compatibility.** Fee-on-transfer and rebasing tokens require adjusted accounting.
3. **Governance-free operation.** Floor-raise scheduling requires authorized invocation; automation via keepers or time-locked schedules can reduce this dependency.

6.11 Future Directions

1. **Extended formal verification**: full contract-level multi-operation invariant preservation across the complete protocol state machine.
2. **Repayment-rate sensitivity**: quantifying how floor appreciation velocity depends on active vs. dormant loan ratios.
3. **Secondary-market convergence**: protocol-owned liquidity positions to tighten convergence between secondary prices and the floor.
4. **Cross-market MEV characterization**: precise analysis of value leakage between primary and secondary venues under various fee regimes.

7 Conclusion

Debt-aware bonding curves offer a fundamentally different approach to DeFi lending. By coupling token issuance with a deterministic, monotonically non-decreasing floor price, the mechanism eliminates the need for liquidation. Borrowing capacity is anchored to a quantity that, by construction, never declines—ensuring that no loan becomes under-collateralized through market movements.

The key properties—floor monotonicity, reserve solvency, and non-liquidatable loan safety—are not policy choices or governance parameters. They are structural consequences of the mechanism's mathematics, proved in the companion paper [Ömer Demirel et al., 2026] and verified at the implementation level through stateful fuzz testing and formal verification (24 Certora CVL rules, 8 Halmos properties).

For token issuers, the mechanism provides a self-reinforcing value accrual loop: trading fees and origination fees fund floor appreciation, which improves capital efficiency for all participants. For borrowers, it provides leverage without liquidation risk. For the broader DeFi ecosystem, it demonstrates that for issuance-native assets, lending need not rely on liquidation cascades as the fundamental solvency mechanism.

The guarantees are specific to this setting: they do not extend to lending against external assets, and floor appreciation remains contingent on fee revenue or other collateral injection.

For formal proofs, see the companion academic paper [Ömer Demirel et al., 2026]. Verification artifacts are available at `https://github.com/demirelo/dabc-proofs`.

References

Aave. Aave protocol whitepaper, version 1.0. Whitepaper, 2020. URL https://github.com/aave/aave-protocol/blob/master/docs/Aave_Protocol_Whitepaper_v1_0.pdf.

Sirio Aramonte, Wenqian Huang, and Andreas Schrimpf. DeFi risks and the decentralisa-

tion illusion. *BIS Quarterly Review*, 2022. URL https://www.bis.org/publ/qtrpdf/r_qt2112b.htm.

Baseline Markets. Baseline protocol: Automated tokenomics engine. Protocol Documentation, 2024. URL https://docs.baseline.markets/.

Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *IEEE Symposium on Security and Privacy (SP)*, 2020. doi: 10.1109/SP40000.2020.00040.

Ömer Demirel. Discrete bonding curves. Medium, 2023a. URL https://medium.com/@demirelo/piecewise-constant-bonding-curves-fcc826449acd.

Ömer Demirel. Dynamic discrete bonding curves. Medium, 2023b. URL https://medium.com/@demirelo/dynamic-kpi-bonding-curves-55b3bf5602bc.

Ömer Demirel. Protocol-owned automated market makers. Medium, 2024. URL https://medium.com/@demirelo/protocol-owned-automated-market-makers-0cfdb110f5a3.

Lioba Heimbach and Wenqian Huang. DeFi leverage. Working Paper 1171, Bank for International Settlements, 2024. URL https://www.bis.org/publ/work1171.pdf.

Robert Leshner and Geoffrey Hayes. Compound: The money market protocol. Whitepaper, 2019. URL https://github.com/compound-finance/compound-money-market/blob/master/docs/CompoundWhitepaper.pdf.

MakerDAO. The Maker protocol: MakerDAO's multi-collateral Dai (MCD) system. Whitepaper, 2017. URL https://makerdao.com/whitepaper/DaiDec17WP.pdf.

Nirvana Finance. Nirvana protocol documentation. Protocol Documentation, 2022. URL https://docs.nirvana.finance/.

Olympus DAO. Olympus protocol: OHM is smart money. Protocol Documentation, 2021. URL https://docs.olympusdao.finance/. Includes cadCAD modeling by BlockScience.

Daniel Perez, Sam M. Werner, Jiahua Xu, and Benjamin Livshits. Liquidations: DeFi on a knife-edge. In *Financial Cryptography and Data Security (FC)*, 2021. doi: 10.1007/978-3-662-64331-0_24.

Phoebe Tian and Yu Zhu. Liquidation mechanisms and price impacts in DeFi. Staff Working Paper 2025-12, Bank of Canada, 2025. URL https://www.bankofcanada.ca/2025/03/staff-working-paper-2025-12/.

Ömer Demirel, Michael Lewkowitz, and Tiago Santana. Debt-aware bonding curves: Non-decreasing floor prices and non-liquidatable borrowing. Cryptology ePrint Archive, Paper 2026/483, 2026. URL https://eprint.iacr.org/2026/483.