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

Ömer Demirel[*] Floors Finance

Michael Lewkowitz[†] House Markets

Tiago Santana[‡] Floors Finance

Abstract

Decentralized lending protocols rely on liquidation mechanisms tied to volatile oracle-derived prices, creating cascading systemic risk during market downturns. Bonding curve token issuance provides deterministic, oracle-free pricing but has not been integrated with lending to eliminate liquidation. We introduce *debt-aware discrete bonding curves*—piecewise-linear bonding curves augmented with a distinguished floor segment whose price is monotonically non-decreasing. A reserve invariant couples the curve's virtual collateral supply to outstanding debt, enabling a non-liquidatable credit facility in which borrowing capacity is anchored to the endogenous floor price rather than a market oracle. We prove that the floor price is monotonically non-decreasing under all protocol operations and that every loan remains solvent without liquidation. A recursive buy-lock-borrow-buy loop enables non-liquidatable leveraged positions at any time; token launches are the most compelling application, as looping efficiency is highest when the floor-to-spot gap is smallest. We support the mechanism with invariant-preserving curve reconfiguration and demonstrate its properties through stateful invariant-based fuzz testing of a concrete implementation. Comparative analysis shows that this design eliminates oracle dependency, liquidation cascades, and health-factor monitoring from the lending design space.

1 Introduction

Decentralized lending protocols—Aave [Aave, 2020], Compound [Leshner and Hayes, 2019], MakerDAO [MakerDAO, 2017]—value collateral at oracle-derived market prices and liquidate positions whose collateralization falls below a safety threshold. This architecture creates cascading sell pressure during downturns [Perez et al., 2021, Tian and Zhu, 2025], attracts MEV extraction on liquidation transactions [Daian et al., 2020], and amplifies volatility across protocols [Heimbach and Huang, 2024].

Several protocols have explored floor-price guarantees as alternatives. Nirvana [Nirvana Finance, 2022] implemented a virtual AMM with a ratcheting floor on Solana but was exploited via flash loan oracle manipulation. Baseline [Baseline Markets, 2024] manages Uniswap V3 concentrated liquidity positions to defend a Base Liquidity Value. Olympus [Olympus DAO, 2021] backs OHM with a treasury and automated monetary policy. Each represents a distinct

[*] omer@floors.finance

[†] michael@house.markets

[‡] tiago@floors.finance

Table 1: Taxonomy of floor-price mechanisms.

Protocol	**Floor mechanism**	**Guarantee**	**Lending**
Nirvana (ANA)	Virtual AMM + reserves	Algorithmic	NIRV stablecoin
Baseline (YES)	Uniswap V3 ranges	Range-dependent	Non-liquidatable
Olympus (OHM)	Treasury + RBS	Policy-dependent	None
This work	Discrete curve invariant	Formally proven	Non-liquidatable

approach, but none provides formal guarantees for floor monotonicity combined with integrated lending.

Building on discrete bonding curves (DBC) [Demirel, 2023a,b] and protocol-owned AMMs [Demirel, 2024b,a]—primary-market issuers that mint tokens on purchase and burn them on sale, embedding *token-owned liquidity* directly in the issuance mechanism, unlike secondary-market AMMs that trade pre-existing inventories—we introduce *debt-aware discrete bonding curves* (DABC): piecewise-linear curves with a floor segment whose price is provably non-decreasing. A reserve invariant couples the curve to outstanding debt, enabling a non-liquidatable credit facility. Our contributions:

1. A formal model of debt-aware discrete bonding curves with a monotonic floor segment (Section 3).
2. A reserve invariant ensuring solvency under all operations (Section 3.3).
3. A non-liquidatable credit facility anchored to the floor price, with proof of loan safety (Section 4).
4. A step-absorption algorithm for floor elevation (Section 3.4).
5. Invariant-preserving curve reconfiguration as a general primitive (Section 3.5).
6. Verification through invariant-based fuzz testing and comparative analysis (Section 5).
7. Non-liquidatable leveraged positions: the recursive leverage loop amplifies exposure without liquidation risk at any point in the curve's lifecycle, with token launches as the most compelling use case (Section 4.5).

2 Related Work

Bonding curves and AMMs. Bancor [Hertzog et al., 2017] introduced continuous bonding curves with a connector-weight model. Uniswap V2 [Adams et al., 2020] and V3 [Adams et al., 2021] established constant-product and concentrated-liquidity AMMs for secondary markets. Cartea et al. [Cartea et al., 2024] study strategic bonding curve design in AMMs. Kirste et al. [Kirste et al., 2025] formalized supply-sovereign AMMs with undergirding bonding curves. Our work extends the discrete bonding curve (DBC) [Demirel, 2023a] and dynamic DBC [Demirel, 2023b] families with a debt-aware reserve invariant and formal floor monotonicity.

Floor-price protocols. Table 1 summarizes the design space. None of the existing approaches provides both formal floor monotonicity and integrated lending.

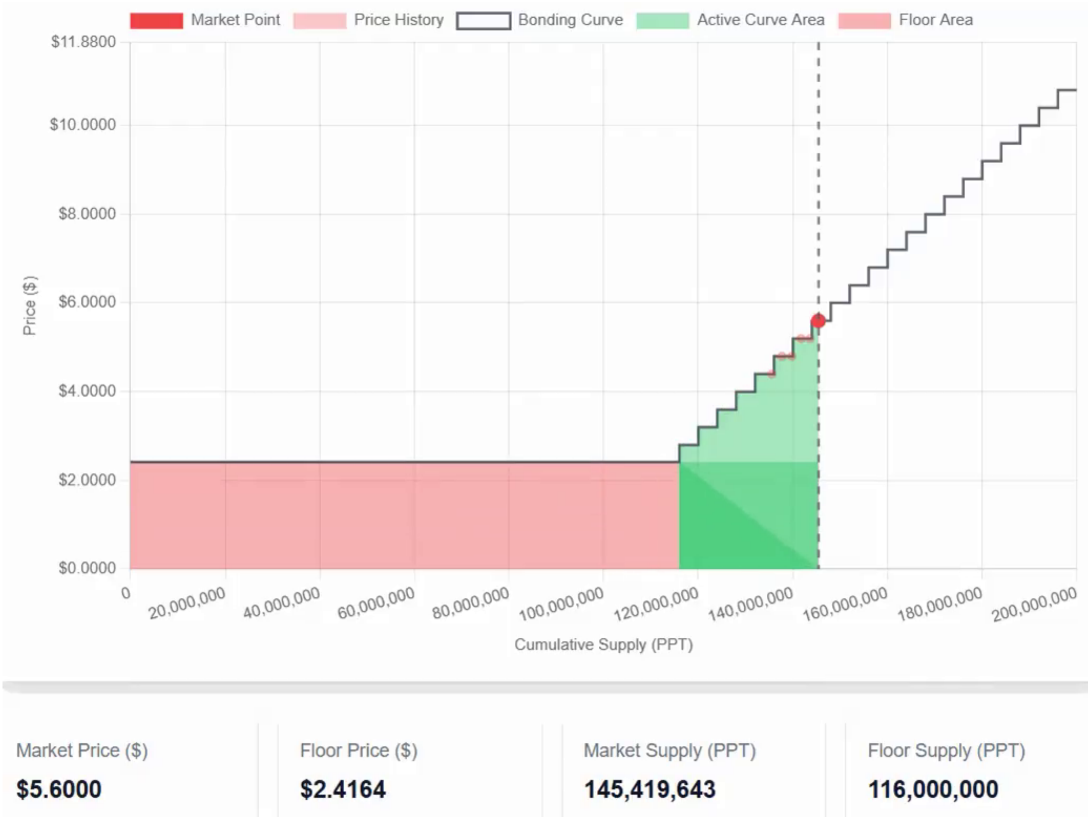

Figure 1: Anatomy of a discrete bonding curve. The floor area (pink) represents tokens redeemable at P_f; the active curve area (green) shows filled premium segments with increasing step prices. The dashed line traces the full curve capacity. The market point (red dot) marks the current supply and price.

DeFi lending. Aave [Aave, 2020], Compound [Leshner and Hayes, 2019], and MakerDAO [MakerDAO, 2017] use oracle-based health factors with liquidation auctions. Curve's crvUSD [Egorov, 2023] introduced soft liquidation via continuous collateral conversion. Timeswap [Timeswap Labs, 2022] and Ajna [Ajna Labs, 2023] remove oracle dependency but retain liquidation or maturity-based default. All existing protocols require liquidation, forced conversion, or default; our mechanism eliminates all three.

3 Model

We formalize the discrete bonding curve mechanism, the reserve invariant, the floor-elevation algorithm, and the reconfiguration principle. Figure 1 illustrates the key components: the flat floor segment (pink), stepped premium segments (green), the current market point, and the full bonding curve outline (dashed).

The bonding curve operates as a primary-market issuer: purchases mint tokens and deposit collateral; sales burn tokens and withdraw collateral. Because the protocol controls the entire supply, the floor price is enforceable through guaranteed redemption.

3.1 Discrete Bonding Curve Segments

Definition 3.1 (Segment)**.** A *segment* is a tuple $\sigma = (p_0, \Delta p, q, n)$, where $p_0 > 0$ is the *initial price*, $\Delta p \geq 0$ the *price increment per step*, $q > 0$ the *supply per step*, and $n > 0$ the *number of steps*. The *price at step* i $(0 \leq i < n)$ is $p(i) = p_0 + i\,\Delta p$, and the *segment capacity* is $C(\sigma) = n \cdot q$. A segment is *flat* when $\Delta p = 0$.

Definition 3.2 (Discrete Bonding Curve). A *discrete bonding curve* is an ordered sequence $\Gamma = (\sigma_0, \sigma_1, \ldots, \sigma_{k-1})$, $k \leq K_{\max}$, satisfying: for every adjacent pair, $p_0^{(i)} + (n^{(i)} - 1)\,\Delta p^{(i)} \leq p_0^{(i+1)}$.

3.2 Floor Segment and Reserve Function

Definition 3.3 (Floor Curve). A curve Γ is a *floor curve* if σ_0 has $n_0 = 1$ and $\Delta p_0 = 0$. The constant $P_f := p_0^{(0)}$ is the *floor price* and $S_0 := q_0$ is the *floor supply*. Every token in the floor segment is redeemable at exactly P_f.

Definition 3.4 (Reserve Function). The *reserve function* $R(\Gamma, S)$ gives the total collateral required to back supply S. For a flat segment: $R_\sigma(s) = \lceil s \cdot p_0/10^{18} \rceil$. For a sloped segment with m full steps and remainder r:

$$R_\sigma(s) = \left\lceil \frac{q \cdot m(2p_0 + (m-1)\Delta p)/2}{10^{18}} \right\rceil + \left\lceil \frac{r(p_0 + m\Delta p)}{10^{18}} \right\rceil.$$

The global reserve iterates over segments: $R(\Gamma, S) = \sum_j R_{\sigma_j}(\min(s_j, C(\sigma_j)))$. All collateral computations round *up* (protocol-favorable); token computations round *down*.

3.3 The Solvency Invariant

Invariant 3.1 (Solvency). At every state transition:

$$V = R(\Gamma, S), \tag{1}$$

where V is the *virtual collateral supply* (an onchain counter tracking total collateral backing), S the total token supply, and Γ the active curve. When the credit facility lends collateral, the actual token balance decreases but V remains unchanged; the locked tokens' floor value covers the difference (since $d_i \leq \gamma \cdot P_f \cdot \ell_i$ and $\gamma < 1$).

Lemma 3.1 (Reserve Lower Bound). *For any floor curve Γ with floor price P_f and any $x \geq 0$: $R(\Gamma, x) \geq \lceil x \cdot P_f/10^{18} \rceil$.*

Proof. Follows from segment price monotonicity (Definition 3.2) and ceiling rounding. □

Proposition 3.1 (Redemption Liquidity). *Let $D = \sum_i d_i$ be total outstanding debt, $A = V - D$ the actual reserve balance, and $S_{\text{free}} = S - \sum_i \ell_i$ the unlocked supply. Locked tokens cannot be sold or transferred; the protocol enforces this through custodial transfer to the credit facility contract at loan creation. Then $A \geq R(\Gamma, S) - R(\Gamma, S - S_{\text{free}})$: actual reserves suffice to pay any sell of unlocked tokens.*

Proof. By Lemma 3.1, $R(\Gamma, S - S_{\text{free}}) \geq \lceil \sum_i \ell_i \cdot P_f/10^{18} \rceil \geq D$ (since $d_i < P_f \cdot \ell_i/10^{18}$ by $\gamma < 1$), giving $A = R(\Gamma, S) - D \geq R(\Gamma, S) - R(\Gamma, S - S_{\text{free}})$. □

Proposition 3.2 (Redemption Preserves Floor). *If x tokens are redeemed at P_f: $(A_f - xP_f)/(S_0 - x) = P_f$, where $A_f = P_f \cdot S_0$.*

Proof. $A'_f/S'_0 = P_f(S_0 - x)/(S_0 - x) = P_f$. With integer rounding, $\lfloor A'_f/S'_0 \rfloor \geq P_f$. □

Algorithm 1 StepAbsorption: Raise the floor

Require: Segments Γ; collateral budget $c > 0$; actual supply S_{actual}
Ensure: New segments Γ'; $P'_f \geq P_f$

$(P_f, S_0) \leftarrow$ floor segment parameters
$S_0 \leftarrow \min(S_0, S_{\text{actual}})$; $\quad$ *capped* $\leftarrow (S_0 = S_{\text{actual}})$ ▷ One-way flag
$j \leftarrow 1$ ▷ Premium segment index
while $c > 0$ **do**
  $S_{\text{eff}} \leftarrow S_0$ ▷ Already capped
  raisable $\leftarrow \lfloor c \cdot 10^{18} / S_{\text{eff}} \rfloor$
  gap $\leftarrow p_j - P_f$
  if *raisable* $<$ *gap* **then** P_f += *raisable*; **break**
  end if
  c −= $\lceil gap \cdot S_{\text{eff}} / 10^{18} \rceil$; $\quad P_f \leftarrow p_j$
  if ¬ *capped* **then** S_0 += q_j; **if** $S_0 \geq S_{\text{actual}}$ **then** $S_0 \leftarrow S_{\text{actual}}$; *capped* $\leftarrow$ *true*
  end if
  $(j, p_j, q_j) \leftarrow$ next premium step (advance segment if exhausted)
end while
$c_{\text{actual}} \leftarrow R(\Gamma', S) - V$ ▷ Exact amount via reserve function
Construct Γ': new floor $\sigma'_0 = (P_f, 0, S_0, 1)$ + remaining segments
return $(\Gamma', c_{\text{actual}})$

3.4 Floor Elevation: Step-Absorption Algorithm

When collateral is injected, the floor is raised by absorbing premium steps.
The algorithm operates in two modes. In *normal mode* ($S_0 < S_{\text{actual}}$), absorbing a step increases both the floor price and the floor supply. In *capped mode* ($S_0 = S_{\text{actual}}$), the floor supply is frozen and only the price increases; the *capped* flag is one-way. After the loop, the implementation recomputes the exact collateral $c_{\text{actual}} = R(\Gamma', S) - V$ via the reserve function, ensuring Invariant 3.1 holds regardless of intermediate rounding.

Harmonic step sizing. Each absorbed step adds its capacity q_j to S_0, making future elevations more expensive (cost $\propto S_0$). With uniform capacities, the cumulative cost scales as $O(m^2)$ in the number of absorbed steps m. Setting $q_j \propto 1/j$ (harmonic sizing) yields $S_0 = O(\log m)$ after m absorptions, reducing the cumulative cost to $O(m \log m)$ (since $S_0 = O(\log m)$ implies per-absorption cost $O(\log m)$, summing to $O(m \log m)$), ensuring long-term floor appreciation remains feasible.

Proposition 3.3 (Floor Elevation Correctness)**.** *Let Γ' be produced by Algorithm 1 with injection c. Then: (a) $P'_f \geq P_f$ (floor price non-decreasing); (b) $S'_0 \geq S_0$ (floor supply non-decreasing); (c) $R(\Gamma', S) = V + c_{\text{actual}}$, where $c_{\text{actual}} \leq c$ is the collateral actually consumed.*

Proof sketch. (a) Every iteration either increases P_f by the full gap to the next step or by a positive residual; neither decreases P_f. (b) S_0 is only modified by addition; the cap bounds the increase. (c) The protocol computes $c_{\text{actual}} = R(\Gamma', S) - V$ exactly via the reserve function and transfers only this amount, ensuring Invariant 3.1 holds after the update. □

3.5 Invariant-Preserving Curve Reconfiguration

Prior work fixed bonding curves at deployment [Demirel, 2023a] or permitted KPI-triggered adjustments [Demirel, 2023b]. We generalize: any reconfiguration is valid provided the reserve invariant holds.

Definition 3.5 (Valid Reconfiguration)**.** Given a governance-configured *maximum spot-loss parameter* $\delta_{\max} \geq 0$, a new curve Γ' is a *valid reconfiguration* iff: (i) $R(\Gamma', S) = V$ (with additional collateral δ: $R(\Gamma', S) = V + \delta$); and (ii) for every supply level $s \leq S$: $P(\Gamma', s) \geq P(\Gamma, s) - \delta_{\max}$. Setting $\delta_{\max} = 0$ makes the reconfiguration fully non-dilutive.

The bounded spot-loss constraint prevents a single reconfiguration from redistributing premium-tier value to the floor. Token holders should be aware that each LRE may decrease the spot price by up to $\delta_{\max}$; this parameter is part of the curve's public configuration. Floor elevation is a special case satisfying both constraints by construction. *Liquidity Reallocation Events (LREs)* are governance-authorized reconfigurations that reshape premium liquidity.

Proposition 3.4 (Reconfiguration Safety)**.** *Any reconfiguration satisfying Definition 3.5 preserves solvency, floor monotonicity (if $P'_f \geq P_f$), and bounds spot-price dilution to at most $\delta_{\max}$.*

This elevates the bonding curve from a static schedule to a *configurable liquidity primitive*.

3.6 Why Discrete Curves?

The choice of a discrete (stepped) bonding curve over a continuous one (e.g., polynomial or exponential) is deliberate:

1. **Deterministic pricing and natural floor.** Within each step, the price is constant (no intra-step slippage). A flat step ($\Delta p = 0$, $n = 1$) maps directly to a guaranteed floor price; continuous curves have no canonical flat region.
2. **Onchain computability.** Reserve computations use only integer arithmetic. No transcendental functions (ln, exp) are needed. Floor elevation reduces to iterating over discrete steps (Algorithm 1).
3. **Composability with lending.** Discrete structure confines rounding to step boundaries, simplifying invariant checks when the credit facility operates.
4. **Formal verifiability.** Discrete state spaces are amenable to exhaustive fuzz testing.
5. **Approximation of continuity.** Sufficiently small step sizes approximate any continuous function while retaining the above advantages.

4 Non-Liquidatable Lending

Token holders can borrow the reserve asset against their holdings. Borrowing capacity is evaluated at the *floor price*, not the market price. Since the floor is monotonically non-decreasing, no liquidation mechanism is required. Borrowers repay voluntarily to unlock; non-repayment results in permanent lock, not forced sale.

4.1 Credit Facility Design

Definition 4.1 (Loan)**.** A *loan* is a tuple $L = (\textit{borrower}, \ell, d, \textit{active})$, where ℓ is the number of locked issuance tokens and d is the outstanding debt.

Definition 4.2 (Borrowing Power)**.** Given ℓ locked tokens, LTV ratio $\gamma_{\text{bps}} \in [1, 10{,}000)$ (equivalently $\gamma = \gamma_{\text{bps}}/10{,}000 \in (0, 1)$), and floor price P_f, the *maximum borrowing power* is

$$\text{maxBorrow}(\ell) = \left\lfloor \frac{\ell \cdot \gamma_{\text{bps}} \cdot P_f}{10^{18} \cdot 10{,}000} \right\rfloor. \tag{2}$$

Borrowing power is computed against P_f, not spot price. The credit facility calls `withdrawCollateralTo` to obtain reserves *without reducing* V, preserving the solvency invariant. The LTV ratio γ_{bps} may be adjusted by governance; lowering it affects only future borrows, not existing loan safety. The one-time origination fee $\phi > 0$ is charged at loan creation; no streaming interest accrues. This is a prerequisite for non-liquidatable safety: streaming interest at rate r would grow $d(t) = d_0 e^{r(t-t_0)}$, requiring $\dot{P}_f/P_f \geq r$ at all times—a bound no protocol can guarantee. Fixing d at origination ensures the ratio only improves as P_f rises. Unlike pooled lending (Aave, Compound), borrowing capacity is *non-rivalrous*: each minter's locked tokens represent collateral they individually contributed, so one borrower's loan does not diminish another's available capacity.

4.2 Floor Price Monotonicity

Theorem 4.1 (Floor Price Monotonicity)**.** *Under the protocol operations* buy*,* sell*,* raiseFloor*, and* reconfigure *(subject to the onchain precondition $P'_f \geq P_f$), the floor price is monotonically non-decreasing: $P_f^{(t+1)} \geq P_f^{(t)}$ for all time steps t.*

Proof. Buy/sell do not modify the floor segment (P_f unchanged; if sell reduces S below S_0, AdjustFloorToSupply sets $S'_0 = S$ while keeping P_f fixed). RaiseFloor only increases P_f (Proposition 3.3(a)). Reconfigure enforces $P'_f \geq P_f$ as an onchain precondition. □

4.3 Non-Liquidatable Safety

Theorem 4.2 (Non-Liquidatable Safety)**.** *An active loan $L = (\textit{borrower}, \ell, d, \textit{true})$ satisfies $d < P_f \cdot \ell/10^{18}$ for all $t \geq t_{\text{creation}}$, regardless of any subsequent changes to γ_{bps}. No liquidation mechanism is required.*

Proof. **Base case:** at origination, $\ell = \lceil d \cdot 10^{18} \cdot 10{,}000/(\gamma_{\text{bps}} \cdot P_f) \rceil$, so $d \leq \text{maxBorrow}(\ell)$ by construction. **Partial repayment:** unlocking $\ell' = \ell - \lfloor r\ell/d \rfloor$, $d' = d - r$ gives $d'/\ell' \leq d/\ell$ (since $\lfloor r\ell/d \rfloor \leq r\ell/d$). **Floor elevation:** since $d < P_f \cdot \ell/10^{18}$ (by $\gamma < 1$) and P_f only increases (Theorem 4.1), the bound strengthens over time. No other events modify (ℓ, d), so loan safety is a global invariant. □

Table 2 contrasts the proposed credit facility with existing lending designs.

4.4 Recursive Leverage

The buy→lock→borrow→buy loop amplifies exposure. With LTV γ and effective fee rate ϕ (origination plus trading fees), the reinvestment fraction is $\eta = \gamma(1 - \phi)$ and net leverage converges to:

Table 2: Comparison of lending designs.

	This work	**Aave/Compound**	**MakerDAO**
Collateral valuation	Floor P_f	Spot oracle	Spot oracle
Liquidation	None	Health-factor	Keeper auction
Oracle dependency	None	External	External
Interest	One-time fee	Variable rate	Stability fee

Table 3: Net leverage $L_{\text{net}} = 1/(1 - \gamma(1 - \phi))$ for selected parameter combinations.

γ (LTV)	ϕ (fee)	$\eta = \gamma(1 - \phi)$	L_{net}
0.80	0.03	0.776	4.46×
0.85	0.03	0.825	5.71×
0.90	0.03	0.873	7.87×
0.90	0.04	0.864	7.35×
0.95	0.03	0.922	12.82×

Proposition 4.1 (Leverage Bound). *The net leverage achievable through recursive looping is*

$$L_{\text{net}} = \sum_{i=0}^{\infty} \eta^i = \frac{1}{1 - \eta} = \frac{1}{1 - \gamma(1 - \phi)}, \tag{3}$$

provided $\eta < 1$, which holds whenever $\gamma < 1$ or $\phi > 0$.

Proof. At loop i, the incremental position is proportional to η^i times the initial deposit. The sum converges to a standard geometric series. The constraint $\eta < 1$ is ensured by the protocol: $\gamma_{\text{bps}} \leq 10{,}000$ and $\phi > 0$. □

Proposition 4.2 (Leverage Decay). *At loop iteration i, the effective reinvestment fraction decays: $\eta_i = \gamma(1 - \phi) \cdot P_f / P_{\text{spot}}^{(i)} \leq \eta_0 = \gamma(1 - \phi)$, because successive purchases traverse higher-priced steps ($P_{\text{spot}}^{(i)} \geq P_f$, monotonically non-decreasing). Thus $L_{\text{realized}} = \sum_{i=0}^{N} \prod_{k=0}^{i} \eta_k < 1/(1 - \eta_0) = L_{\text{net}}$: leveraged supply expansion is algorithmically self-limiting.*

Table 3 shows leverage sensitivity to parameter choices.

Safety. The solvency invariant is verified at every loop iteration; if remaining collateral is insufficient, the loop terminates early. During presale periods, a time-decaying fee multiplier discourages early aggressive looping.

4.5 Application: Leveraged Token Launches

The recursive leverage mechanism enables non-liquidatable leveraged positions at any point in the curve's lifecycle; token launches are the most compelling application, as looping efficiency is highest when the floor-to-spot gap is smallest. In oracle-based DeFi, leveraged launches are liquidation-prone because collateral is valued at the maximally volatile spot price [Perez et al., 2021]. In our mechanism, borrowing capacity is evaluated against P_f: because P_f is non-decreasing (Theorem 4.1) and every loan remains safe (Theorem 4.2), *no loan in the loop can ever become under-collateralized*. The total position converges to $C_0 \cdot L_{\text{net}}/P_f$ tokens. The one-time origination fee doubles as an algorithmic bootstrapping mechanism: each loop iteration generates fees directed to the floor reserve, replacing inflationary token emissions with non-dilutive floor appreciation.

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

Property	**Ours**	**Nirv.**	**Base.**	**Oly.**	**Aave**
Floor enforcement	•	•	•	◦	◦
Guarantee	Inv.	Alg.	Range	Treas.	N/A
Lending	•	△	△	◦	•
Oracle free	•	◦	△	◦	◦
Floor monotonicity	Proven	Claimed	◦	◦	N/A
Reconfigurable	•	◦	◦	◦	N/A

5 Analysis and Evaluation

5.1 Comparative Analysis

Table 4 positions the present work against five systems with floor-price or reserve-backed mechanics.

Our mechanism is the only design that combines a constructive (invariant-verified) floor guarantee with integrated lending and no oracle dependency. The reconfiguration primitive (Definition 3.5) has no analogue in related work.

5.2 Capital Efficiency

LTV is computed against P_f, which is conservative at origination ($P_f \leq P_{\text{spot}}$) but improves over time:

$$\text{LTV}_{\text{eff}}(t_1) = \frac{d}{P_f(t_1) \cdot \ell} \leq \gamma \cdot \frac{P_f(t_0)}{P_f(t_1)} \leq \gamma.$$

Loans become *better* collateralized without borrower action (*passive de-risking*). Redemptions below S_0 trigger AdjustFloorToSupply, reducing S_0 and making future floor raises cheaper (antifragility): $\Delta P_f^{\text{annual}} \approx V_{\text{daily}} \cdot f \cdot \alpha \cdot 365/S_0$.

5.3 Invariant-Based Verification

A concrete Solidity implementation (*Floors*, ∼20,000 LOC, LGPL-3.0) was verified through stateful invariant-based fuzz testing (Foundry). Handler contracts wrap each module and expose fuzzer-callable entry points executing bounded, randomized operation sequences. Five invariants are checked after every sequence: (1) *reserve solvency* ($|V - R(S)| \leq 100$ wei); (2) *floor monotonicity* ($P_f(t) \geq P_f(0)$); (3) *segment validity* (≥ 2 segments, floor has one flat step); (4) *collateral accounting* (no collateral created or destroyed); (5) *loan safety* ($d_i \leq \gamma \cdot P_f \cdot \ell_i / 10^{18}$ for every active loan). A full-protocol handler orchestrates multi-step scenarios under `fail_on_revert = true`.

5.4 Failure Modes and Trust Assumptions

Under correct implementation, floor monotonicity, reserve solvency, and loan safety are maintained by construction (Theorems 4.1–4.2). Risks include: (i) smart contract bugs (mitigated by fuzz testing), (ii) administrator inaction on floor raises (see Section 6), and (iii) nonstandard collateral tokens (an implementation constraint). The mechanism additionally assumes blockchain liveness and finality.

6 Discussion and Limitations

Administrative trust. The `raiseFloor` operation is permissioned. Onchain preconditions prevent lowering the floor; the administrator controls only the *pace* of elevation, not its direction. Automated, governance-controlled floor raises can further reduce this trust surface. Reconfiguration is bounded by $\delta_{\max}$ (Definition 3.5): each LRE may reduce spot prices by at most this amount. Setting $\delta_{\max} = 0$ restricts reshaping to unminted supply. Existing holders' floor guarantees and lending positions are unaffected in all cases.

Primary-market-only guarantee. The floor applies to the bonding curve's primary market. Secondary markets may temporarily trade below P_f; rational arbitrageurs buying below P_f and redeeming onchain exert convergence pressure. Convergence speed depends on transaction costs and redemption limits.

Fee-dependent sustainability. The floor rises only when trading and credit activity generates fee revenue allocated to the floor reserve. In sustained low-volume periods, floor appreciation slows—though the floor never decreases. Denominating the reserve in yield-bearing assets (e.g., staked ETH) provides a baseline source of floor appreciation independent of trading volume (using liquid staking tokens to preserve instant redeemability).

Other constraints. The segment count $K_{\max}$ bounds curve expressiveness (gas cost scales linearly). The current implementation assumes standard ERC-20 collateral; this is an implementation constraint that does not affect the mechanism's formal properties.

Open-ended debt. The one-time origination fee ϕ with no streaming interest creates open-ended loans. Scaling ϕ with per-borrower utilization (increasing as debt approaches the LTV limit) can discourage over-concentration without compromising the non-liquidatable guarantee, e.g., $\phi(u) = \phi_{\min} + (\phi_{\max} - \phi_{\min})u^2$ where $u = d/\operatorname{maxBorrow}(\ell)$.

MEV at step boundaries. Discrete steps create predictable price transitions that can be front-run. A sufficient condition for sandwich unprofitability is $\Delta p < 2\, f\, p_j$, where f is the round-trip fee rate; curve designers should parameterize Δp and f jointly.

Capital efficiency trade-off. Anchoring LTV to P_f rather than the spot price yields a conservative effective LTV at origination; this is the cost of eliminating liquidation. As P_f rises, capital efficiency improves passively—unlike oracle-based systems where efficiency degrades with volatility.

Leveraged positions and future work. The recursive leverage loop (Section 4.5) constitutes a new primitive: non-liquidatable leveraged positions at any point in the curve's lifecycle. The loop introduces *reflexivity*: leveraged buying generates fees that raise the floor, improving capital efficiency for subsequent loops. Future work includes formal verification (e.g., Certora), cross-chain reserve consistency, automated floor raises, and dynamic fee multipliers that modulate loop intensity based on the premium-to-floor ratio.

7 Conclusion

We presented debt-aware discrete bonding curves, augmenting piecewise-linear bonding curves with a monotonically non-decreasing floor segment and a reserve invariant accounting for outstanding debt. The key insight is that anchoring LTV to an endogenous, monotonic floor price eliminates liquidation: borrowing capacity never decreases, so no loan becomes undercollateralized through market movements.

We proved floor monotonicity (Theorem 4.1) and non-liquidatable safety (Theorem 4.2), and introduced invariant-preserving curve reconfiguration as a new primitive for adaptive market making. Comparative analysis shows the mechanism is the first in our survey to combine floor enforcement with integrated lending while eliminating oracle dependency and liquidation cascades. More broadly, the work demonstrates that DeFi lending can be designed without liquidation as the fundamental solvency mechanism.

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.

Hayden Adams, Noah Zinsmeister, and Dan Robinson. Uniswap v2 core. Whitepaper, 2020. URL https://docs.uniswap.org/whitepaper.pdf.

Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. Uniswap v3 core. Whitepaper, 2021. URL https://app.uniswap.org/whitepaper-v3.pdf.

Ajna Labs. Ajna protocol: Peer-to-pool lending without oracles. Whitepaper, 2023. URL https://www.ajna.finance/pdf/Ajna_Protocol_Whitepaper_01-11-2024.pdf.

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

Álvaro Cartea, Fayçal Drissi, Leandro Sánchez-Betancourt, David Siska, and Lukasz Szpruch. Strategic bonding curves in automated market makers. *SSRN Electronic Journal*, 2024. doi: 10.2139/ssrn.5018420.

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.

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

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

Omer Demirel. Bonding curves for L1 token economies. Medium, 2024a. URL https://medium.com/@demirelo/bonding-curves-for-l1-token-economies-7afb60f392e1.

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

Michael Egorov. crvUSD: Llamma and stablecoin design. Whitepaper, 2023. URL https://github.com/curvefi/curve-stablecoin/blob/master/doc/curve-stablecoin.pdf.

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

Eyal Hertzog, Guy Benartzi, and Galia Benartzi. Bancor protocol: Continuous liquidity for cryptographic tokens through their smart contracts. Whitepaper, 2017. URL https://resources.bancor.network/pages/BancorProtocolWhitepaper.pdf.

Dennis Kirste, Alexander Poddey, and Ali Sunyaev. Undergirding bonding curves: Supply-sovereign automated market makers enabling liquidity and sustainable financing. *Distributed Ledger Technologies: Research and Practice*, 2025. doi: 10.1145/3716177.

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/.

Timeswap Labs. Timeswap: Permissionless, oracle-less, non-liquidatable fixed maturity lending & borrowing. Whitepaper, 2022. URL https://timeswap.io/whitepaper.pdf.