# Static Per-Output Miner Fee

Josh · Secure Sovereign

Pre-Proposal · Standards Track · Created July 23, 2026

CONTENTS



## Abstract

This BIP proposes a consensus-enforced per-output miner fee: a fixed satoshi floor that every non-coinbase transaction must leave as part of its ordinary transaction fee for each output it creates.

Bitcoin has no consensus-enforced fee floor today. Policy-based dust limits and mempool filters are unenforceable, because any transaction that pays enough miner fees can still be included by a participant who chooses not to enforce those filters. This proposal prices the permanent cost of UTXO creation at consensus, and it binds every participant.

The fee is a single fixed constant. There is no dynamic component, no sampling window, no moving average, and no rate adjustment. Every non-coinbase output pays the same fee regardless of script type, value, or fee environment. The amount is paid as ordinary miner fee (`sum(inputs) - sum(outputs)`), not minted into the coinbase. It is not recoverable by the sender. Old nodes see a valid high-fee transaction. New nodes reject transactions that miss the floor. That is a soft fork.

The design is deliberately small. A companion BIP adds a dynamic escalation layer on top of this static fee once the network has operational experience with the base rule. Dedicated embedding channels are a different surface; they are closed by Permanent Data Channel Closure. The three pre-proposals are one stack. This BIP prices the UTXO slot.

## Motivation

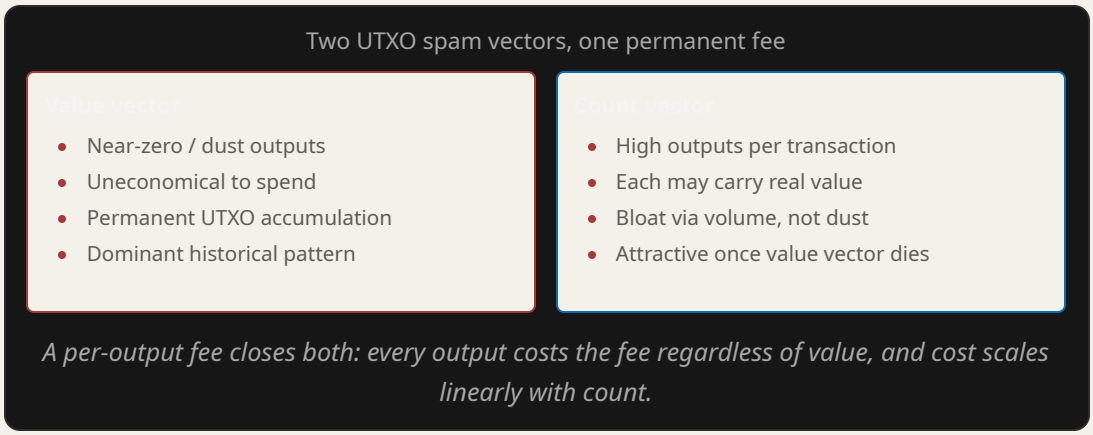
### The Problem: No Consensus Fee Floor Exists

Bitcoin has no consensus rule requiring any minimum fee for transaction inclusion. Policy filters, dust limits, and minimum relay fees can all be bypassed by routing through a miner who does not enforce them. That is why UTXO spam debates have cycled for years without resolution. Policy binds willing participants. Consensus binds everyone.

This BIP introduces a consensus fee floor scoped to UTXO creation: the surface where unpriced externalities are measurable, permanent, and harmful to network health.

### The Two Spam Vectors

UTXO set spam runs on two vectors.




The **value vector** is the main historical pattern: outputs created at or near zero value that are uneconomical to spend and sit permanently in the UTXO set. They cost almost nothing to create because their value is tiny and miner fees can be spread across many outputs.

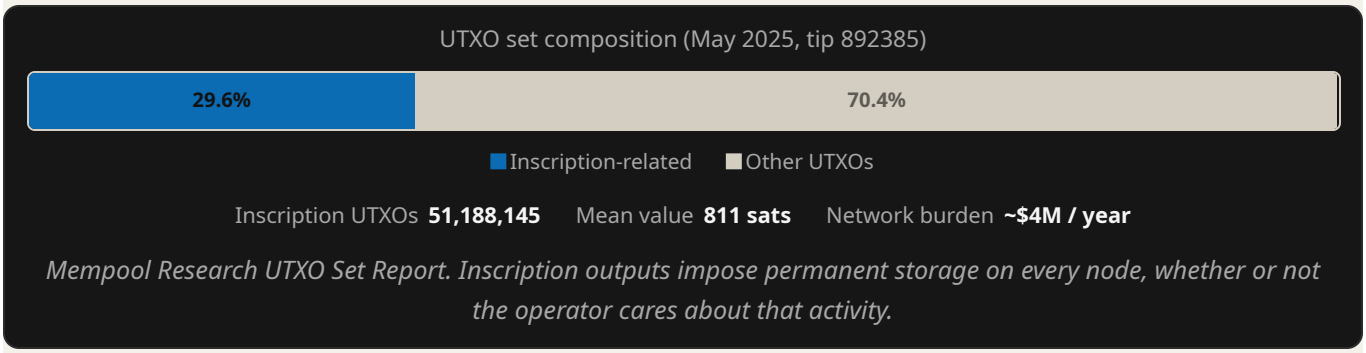
The **count vector** is the secondary pattern: many outputs per transaction, each carrying real value, bloating the UTXO set through volume rather than dust. This vector becomes more attractive once the value vector is closed.

A permanent per-output fee closes both at once. Any output costs the fee to create, whatever its value. The fee scales linearly with output count. There is no capital recovery path: the fee goes to miners and does not come back.

### The Externalized Cost This Proposal Prices

The per-output fee is not a new tax on Bitcoin use. It prices a cost that already falls on node operators instead of the actors who create it.

*Full Cost of Running a Bitcoin Node* (v2.4, July 2026) estimates that burden:




*Mempool Research UTXO Set Report. Inscription outputs impose permanent storage on every node, whether or not the operator cares about that activity.*

| EXTERNALITY | ESTIMATE | WHO PAYS |
|---|---|---|
| **Non-monetary chain share** | 12-19% (~85-140 GB of 700-750 GB) | Every new node at IBD |
| **Annual IBD from inscriptions** | 765 GB - 1.68 TB across ~9-12k new nodes | New participants |
| **Per-node ongoing cost** | $5.52-$5.54 / month (2026) | ~60,000 full nodes |
| **Aggregate network burden** | ~$4M / year | Node operators |
| **Core development spend** | ~$9M / year | (comparison only) |

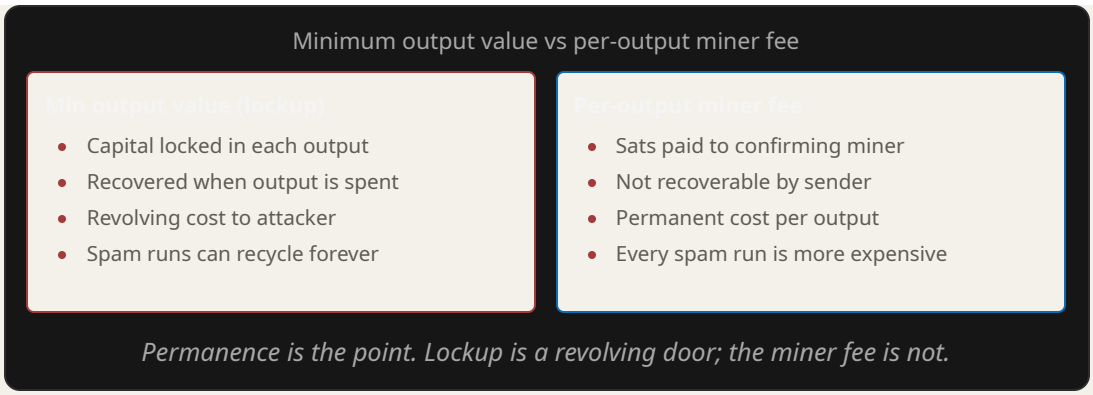
*Node operators absorb a non-monetary burden approaching half of total protocol development spend, with no recovery path.*

That also names a political constituency: about 60,000 node operators absorbing roughly $4 million per year in non-monetary burden. They have a direct interest in this rule. They are not the BIP-8/9 signaling threshold.

### Why Determined Spammers Require a Permanent Fee

The 2023 to 2026 inscription and ordinals waves showed that determined spammers are more price-insensitive than simple models predict. Those actors paid sustained high miner fees because presence on Bitcoin itself was part of the product. Blocks ran 91 to 97% full for long stretches, and the activity continued.

A capital lockup does not change that math. An attacker facing only a minimum output value can mint outputs, lock capital, spend them back, and repeat. The per-output fee removes that path. Every spam run is permanently more expensive, with no revolving capital to fund the next one.




### Provisional Static Fee Anchor

The static fee needs a principled starting point for calibration, not an arbitrary constant. The node-cost work above helps, but the derivation has to ask the right question.

The $4 million per year aggregate and the $5.52 to $5.54 per node per month figures are the ongoing cost of non-monetary data already in the chain: storage, bandwidth, and RAM that existing inscription UTXOs impose every month. That is a stock cost, not a per-new-output flow cost. You cannot divide it by new UTXO creation rates and call the result a fee.

The right question is: what lifetime cost does one newly created non-monetary UTXO impose on the network from creation forward? That cost has three parts:

1. **Permanent storage cost:** about 85 to 140GB of non-monetary chain data at $0.11/GB NVMe pricing is $9 to $15 per node in sunk storage. Spread across about 51 million inscription UTXOs, that is roughly 0.1 to 0.3 sat per UTXO in storage cost per node.
2. **Ongoing monthly storage and RAM burden:** $5.52 per month per node in non-monetary ongoing cost. At 60,000 nodes and 51 million inscription UTXOs, that is about 78 sats per UTXO per year across the network, or roughly 6 to 7 sats per UTXO per node-year.
3. **IBD cost** on every future node that syncs the chain: 85 to 140GB of inscription data downloaded once per new node, at about 9,000 to 12,000 new nodes per year.

Summing those pieces and discounting over a reasonable UTXO lifetime at current BTC price yields a per-UTXO externalized lifetime cost around **16 to 20 sats per output**. That is a cost-derived provisional anchor for the **UTXO-slot** externality, not a conclusion, and not an inscription-killer. The 2023 to 2026 inscription waves already paid far more than 20 sats per output in weight fees. Those waves would have continued at this floor on fee grounds alone. Closing dedicated data channels is the job of *Permanent Data Channel Closure*. This BIP prices output count.

The 16 to 20 sat band must still be checked against two tests: what floor deters high-count UTXO spam at realistic attacker budgets, and what floor stays negligible relative to legitimate output values across historical fee regimes. If calibration shows 16 to 20 sats is too low to deter determined count-vector spam, the static fee will be raised and the change documented before

the proposal moves forward. If the floor is not negligible for Lightning channel opens, CoinJoin outputs, and exchange batch withdrawals, the static fee will be lowered and the change documented the same way.

## Specification

### Fee Rule

This is a transaction-fee floor, not extra mint. Coinbase accounting is unchanged from existing consensus (`coinbase_value <= block_subsidy + sum(tx_fees)`). A block of transactions that each meet the per-output floor can still have a miscounted coinbase, so that existing check remains required in addition to the per-transaction floor. The floor is a consensus validity rule, not relay policy. Miner preference cannot override it. Adding the per-output amount to the coinbase above subsidy-plus-fees would be a hard fork and is not this BIP.

For every non-coinbase transaction in a block at or above `activation_height`:

```
tx_fee = sum(input_values) - sum(output_values)
n_outputs = count(outputs in this transaction)
tx_fee >= active_fee × n_outputs
```

```
active_fee = static_fee
```

`tx_fee` is the existing consensus fee: a non-negative integer satoshi amount. Transactions with `sum(input_values) < sum(output_values)` remain invalid under existing rules. `static_fee` is a consensus constant fixed at activation. It changes only by a future soft fork.

Overpayment is valid. The floor is per transaction from that transaction's own inputs and outputs. Package fee rate and CPFP do not count: a parent that misses the floor is invalid even if a child pays more.

All arithmetic is integer satoshis. Multiplication uses at least 64-bit unsigned range. A transaction that overflows that range is invalid.

### Exempt Outputs

The coinbase transaction is the only exemption. Charging coinbase outputs would be circular: the coinbase is how the miner collects `tx_fee`. All other transactions pay the floor, including those whose outputs are `OP_RETURN` or otherwise unspendable.

### Validation

1. Skip the coinbase transaction.
2. For each remaining transaction, compute `tx_fee` and `n_outputs` as above. A transaction with zero outputs has `n_outputs = 0` and meets the floor automatically.
3. If `tx_fee < static_fee × n_outputs`, the transaction is invalid and the block is invalid.
4. Existing coinbase rules are unchanged. Miners receive the per-output amount because it is part of `sum(tx_fees)`.

This is a soft fork. Old nodes accept any non-negative fee. New nodes reject transactions that miss the floor. A miner who includes a non-compliant transaction produces a block that enforcing nodes reject.

### Activation Height

The fee rule applies to every block at or above `activation_height`. Blocks below `activation_height` are unaffected. `activation_height` is set by the BIP-8 or BIP-9 signaling process and is known in advance, so wallets and services have time to update fee estimation. No separate grace window is specified: the minimum one-year signaling window is enough protection for in-flight transactions, and an extra grace window adds complexity without much added safety.

### Consensus State

The only consensus state this BIP requires is:

- `static_fee` (fixed at activation)
- `activation_height` (set by the signaling process)

Both are deterministic from the soft fork activation parameters and need no ongoing computation.

### Parameters

| PARAMETER | DESCRIPTION | PROVISIONAL VALUE |
|---|---|---|
| **`static_fee`** | Fixed per-output fee in sats | 16-20 sats (pending calibration) |

At the provisional band, count-vector cost scales linearly:

| OUTPUTS CREATED | @ 16 SATS | @ 20 SATS |
|---|---|---|
| **1,000** | 16,000 sats | 20,000 sats |
| **10,000** | 160,000 sats | 200,000 sats |
| **100,000** | 1,600,000 sats | 2,000,000 sats |

*Permanent, non-recoverable cost per spam run. Calibration must confirm deterrence at realistic attacker budgets and negligibility for legitimate high-output use.*

*Permanent, non-recoverable cost per spam run. Calibration must confirm deterrence at realistic attacker budgets and negligibility for legitimate high-output use.*

### Activation

This BIP is meant to deploy via a soft fork using BIP-8 or BIP-9 style signaling, with a minimum activation window of one year and no mandatory lock-in fallback. Miners who do not signal are not penalized. If signaling does not reach the threshold within the window, the proposal does not activate and the process restarts with revised parameters or renewed community discussion.

BIP-8/9 without lock-in-on-timeout is miner signaling. Node operators absorbing non-monetary burden are a political constituency for this rule; they are not the signaling threshold. If miners do not signal, this BIP does not activate under the specified mechanism. User-activated soft fork deployment is a separate choice and is not specified here.

## Rationale

### Why a Fee Floor, Not Extra Mint

The per-output amount must be part of `tx_fee`. Minting it on top of subsidy-plus-fees would raise `coinbase_value` above what pre-activation nodes allow, which is a hard fork. A floor on `tx_fee` is a soft fork: old nodes accept the transaction; new nodes reject a miss. Miners still receive the sats. The amount is still permanent and not recoverable by the sender.

### Why a Permanent Fee Rather Than a Capital Lockup

A minimum output value floor forces attackers to lock capital in each output. That capital returns when the output is spent, so the floor is a revolving cost. The per-output miner fee removes that path: those sats go to miners and do not return. Every spam run is permanently more expensive.

### Why this differs from prior minimum fee proposals

Prior minimum fee proposals were either relay policy, which miners can bypass, or minimum output value floors, which return locked capital when outputs are spent. This proposal is neither. The per-transaction consensus validity rule means a non-compliant transaction cannot appear in any valid block, regardless of miner preference or side arrangements. The fee is paid at confirmation and is not recoverable by the sender.

### Why a Single Global Rate With No Script-Type Differentiation

A single constant per output is the simplest rule. Script-type differentiation would force the fee to track script classification across every output form, create edge cases at upgrade boundaries, and make the rule harder to specify cleanly. The externalized cost this fee prices is per UTXO slot, not per script type, so a uniform fee matches the cost model.

### Why OP_RETURN Pays the Fee

All non-coinbase outputs pay the fee, including `OP_RETURN` and provably unspendable outputs. Legitimate `OP_RETURN` users will object: timestamping services, colored coin protocols, and apps that embed small amounts of data for non-spam purposes.

The tradeoff is accepted for two reasons. First, the motivation is the cost imposed on node operators by all non-coinbase output creation, including outputs that never enter the UTXO set. `OP_RETURN` outputs consume block space and impose bandwidth and storage costs on every node whether or not they are spendable. Exempting them while citing node operator burden would contradict the premise. Second, at a correctly calibrated static fee of 16 to 20 sats, the cost per `OP_RETURN` output is negligible for legitimate low-volume use and material only for high-volume data embedding, which is the intended effect.

### Why a Static-Only Rule First

A combined static-plus-dynamic rule has more review surface, more implementation complexity, and more political attack surface than a static-only rule. The property that matters most is that a permanent, non-recoverable cost exists on every new output. That property lives entirely in the static component. The dynamic escalation layer is a long-term anti-decay mechanism that is useful but not urgent. Staging the two rules raises the odds that the core mechanism actually reaches consensus.

### Miner Incentives

Miners receive per-output fees as part of ordinary `tx_fee` in the coinbase. They have a financial interest in including transactions that create outputs. That is not new: miners already want fee-paying transactions. The per-output fee raises the price floor on all output creation. Spammers pay miners more per output than before. The per-output cost to attackers rises whether or not aggregate spam volume falls.

Large miners who earn significant revenue from inscription-style activity may oppose this. That opposition is short-term: a fee market where monetary transactions compete on equal terms with correctly priced non-monetary use produces more durable revenue than one distorted by externalized costs.

### Interaction with Permanent Data Channel Closure

The Permanent Data Channel Closure pre-proposal targets dedicated high-bandwidth data channels: `OP_RETURN`, Taproot envelopes, witness fragmentation, and related embedding surfaces. This BIP targets UTXO creation economics and charges all outputs, including `OP_RETURN`. The two proposals address different surfaces and reinforce each other when activated together. Coordinated activation of all three pre-proposals closes dedicated embedding channels and prices UTXO creation. Hash, amount, `nSequence`, and ordering fields remain open because payments require them; see *The Achievable Floor* and *Bitcoin Is Not a Hard Drive*.

### Activation Game Theory

Prior soft fork proposals that use mandatory lock-in fallbacks treat non-signaling miners as attackers rather than participants with legitimate concerns. This BIP uses voluntary signaling with no mandatory lock-in. If inscription-dependent miners block signaling, that outcome documents the network's governance dynamics and strengthens the case for implementation diversity and alternative node software. It does not activate the rule. Node-operator interest is not a substitute for the specified signaling threshold.

The most credible activation path is to deploy on signet and testnet, run the rule in production across Bitcoin Commons and Bitcoin Knots, publish calibrated parameters with full chain scan results, and treat miner signaling as a multi-year process.

## Backwards Compatibility

Existing UTXOs remain valid and spendable. Old nodes will see new blocks as valid under the soft fork. From the user's point of view there is one fee: wallets add the weight-based transaction fee and the per-output component internally and show a single total. Wallets that do not update will underestimate fees and produce transactions that enforcing nodes reject at and after the activation height. The one-year minimum signaling window gives the wallet ecosystem time to update. Updates should be complete before activation: unlike soft forks that affect only unusual transaction types, the per-output fee affects fee estimation for nearly every transaction that creates outputs. A wallet that is not updated by activation will start producing invalid transactions the moment the rule applies, with no warning to the user.

Lightning channel opens, CoinJoin transactions, and exchange batch payouts will pay the per-output fee for each output created. Calibration must confirm the fee is negligible relative to typical output values across historical fee regimes. Exchanges should be engaged before signaling begins.

## Reference Implementation

High-level pseudocode:

```
STATIC_FEE = <value to be set at activation>
ACTIVATION_HEIGHT = <determined by signaling process>

def tx_fee(tx):
    return sum(inp.value for inp in tx.inputs) - sum(out.value for out in tx.outputs)

def is_valid_transaction(tx, block_height):
    # Transaction validation, before block assembly.
    # Per-transaction floor: no tx can free-ride on another tx's fees in the same block.
    if block_height < ACTIVATION_HEIGHT or tx.is_coinbase:
        return True
    return tx_fee(tx) >= STATIC_FEE * len(tx.outputs)

def is_valid_block(block):
    # First guard: every transaction must already meet the per-output floor.
    if not all(is_valid_transaction(tx, block.height) for tx in block.transactions):
        return False
    # Second independent guard: existing coinbase accounting.
    fees = sum(tx_fee(tx) for tx in block.transactions if not tx.is_coinbase)
    return coinbase_value(block) <= block_subsidy(block.height) + fees
```

The per-transaction function does not check coinbase value. `is_valid_block` applies existing subsidy-plus-fee rules as a second independent guard.

Detailed test vectors, integer arithmetic precision requirements, and treatment of edge cases (zero-output transactions, reorgs at activation height, IBD validation) will be provided in a future numbered BIP submission.

## Calibration Checklist

Calibration is a hard gate. The static fee value is provisional until this checklist is satisfied.

### A. Static Fee Determination (Hard Gate)

- Validate the provisional 16-20 sat anchor as a UTXO-slot price: stress-test against high-count attack economics at realistic attacker budgets (1,000 / 10,000 / 100,000 outputs). Raise the static fee if the provisional number does not deter count-vector spam at scale. Do not treat "would inscriptions have stopped at 20 sats" as a pass condition; that is the data-channel BIP.
- Confirm the static fee is negligible relative to Lightning channel open values, CoinJoin outputs, and exchange batch withdrawal amounts at the 99th percentile fee rate.
- Confirm the static fee does not make ordinary small payment outputs uneconomical under any historical fee regime.
- Lock in the static fee value, or document the required adjustment with full reasoning.

### B. Collateral Damage (Must Pass)

- Lightning channel opens: per-output fee as a percentage of typical funding output value; pass if negligible across historical fee regimes.
- Exchange batch withdrawals: total per-output fee burden as a percentage of transaction value; pass if negligible.
- CoinJoin / JoinMarket / Wasabi-class transactions: same analysis.
- `OP_RETURN`-bearing transactions: confirm the per-output fee does not break legitimate low-volume `OP_RETURN` use.
- Ordinary payments: per-output fee near zero relative to output value.

### C. Spam Efficacy

- Verify the static fee makes the documented value-vector spam pattern permanently uneconomical.
- Confirm no capital recovery path exists under the per-output fee mechanism.
- Model inscription-wave attacker economics **together with** the data-channel BIP: the static fee alone does not claim to have stopped 2023-2026 dedicated embedding.

### D. Consensus Edge Cases

- Confirm reorg behavior at activation height: blocks below `activation_height` are never subject to the fee rule, regardless of reorg depth.
- Confirm IBD and assumeutxo: activation height and static fee are reconstructable from soft fork parameters alone.
- Confirm integer arithmetic: satoshi amounts, 64-bit multiply of `static_fee × n_outputs`.
- Confirm coinbase exemption: the floor applies to non-coinbase transactions only; coinbase value uses existing subsidy-plus-fee rules.
- Confirm the floor is per-transaction `tx_fee`, not package or CPFP attribution.

### E. Wallet and Exchange Validation

- Share calibration results with at least one exchange wallet engineer and one Lightning implementation developer before proceeding.
- Confirm wallet fee estimation correctly sums weight-based fee and per-output fee into a single user-facing total.
- Confirm fee estimation APIs can be updated without breaking existing interfaces.

### F. Exit Criteria

- Static fee value confirmed and gate A passes, including the high-count stress test.
- Collateral checklist B passes at the recommended static fee.
- Spam efficacy checklist C confirms permanent cost on both vectors.
- Consensus edge cases in checklist D resolved and written into Specification.

## Security Considerations

**No dynamic manipulation surface.** The static fee is a constant. It has no sampling window, no moving average, and no parameter an attacker can influence through fee activity. That is the main security advantage of the static-only design.

**Soft fork boundary.** The floor restricts currently valid (low-fee) transactions. It does not mint coins. Blocks that meet the floor are valid to old nodes.

**Insufficient tx fee.** A transaction with `tx_fee < static_fee × n_outputs` is invalid. A block that includes it is invalid. Self-enforcing among enforcing nodes. CPFP cannot rescue the parent.

**Clean activation boundary.** The rule takes effect at `activation_height` with no grace window. The one-year minimum signaling window is enough for wallets and services to update before the rule applies. Transactions broadcast before `activation_height` that remain unconfirmed at that height must meet the floor to confirm. Wallets should watch approaching activation and rebroadcast or bump fees as needed.

**Residual incentive surfaces.** A transaction that misses the per-output floor is consensus-invalid. Including it does not produce an orphan; it produces a block that cannot exist. No side payment can make that transaction valid.

A miner can still accept the full fee and rebate part of it to the spammer after confirmation. That requires a sustained operational arrangement. The miner absorbs the rebate as a direct cost to coinbase revenue. Publicly broadcast transactions can be mined by anyone, so the spammer's effective throughput is limited by that miner's hashrate share.

A vertically integrated spammer-miner faces the same hashrate-share cap on self-inclusion and pays full fees when submitting publicly.

These are known residual vectors. The dynamic escalation BIP and Permanent Data Channel Closure address surfaces this proposal does not close. The three pre-proposals are complementary, not redundant.

**Static fee decay over time.** A fixed satoshi amount becomes economically trivial as BTC price and fee levels rise over decades. That is the known limit of the static-only design and the reason for the companion dynamic escalation BIP. The static fee is not meant as a permanent final answer; it is the first layer the dynamic BIP builds on.

## References


- *BIP Pre-Proposal: Dynamic Escalation of the Per-Output Miner Fee* (Josh / Secure Sovereign, July 2026). Optional companion.
- *BIP Pre-Proposal: Permanent Data Channel Closure* (Josh / Secure Sovereign, August 2026).
- Bitcoin Core dust limit implementation ( `GetDustThreshold` / `-dustrelayfee` ; Core PRs #2577, #10817, #22863)
- *Full Cost of Running a Bitcoin Node*, v2.4, July 2026. Source of per-node non-monetary burden ($5.52-$5.54/month) and network aggregate ($4M/year) figures.
- Mempool Research UTXO Set Report, May 2025, tip 892385 (research.mempool.space/utxo-set-report). Source of 29.6% inscription UTXO share and 51,188,145 inscription UTXO count.
- *The Achievable Floor* (2026).
- *Bitcoin Is Not a Hard Drive* (2026).
- Lopp, Jameson. "Goldiblocks: A Dynamic Block Size Limit." OP_NEXT 2025.
- Various Delving Bitcoin and bitcoin-dev threads on UTXO bloat and spam mitigation (2023-2026)


## Copyright



## RELATED

Related BIP Dynamic Escalation of the Per-Output Miner Fee

Works with Permanent Data Channel Closure

Related writing

Full Cost of Running a Bitcoin Node · The Achievable Floor · Bitcoin Is Not a Hard Drive

**SECURESOVEREIGN**

Markdown is the canonical source: https://secsov.com/bips/static-per-output-miner-fee/index.md

Formatted snapshot of https://secsov.com/bips/static-per-output-miner-fee

Josh · Secure Sovereign