Appendix A
Technical Details

More technical details of Section III are provided below.

A. *Adaptive Pipeline Scheduling*

APS. First is the triple-decoupling. ❶ For decoupling broadcast from consensus: After preparing a proposal, the primary first enqueues the proposal into a broadcast queue. A dedicated thread then handles broadcasting to all replicas, allowing the primary to proceed without blocking on the broadcast. Broadcast messages are pre-validated to ensure correctness. This step can be completed in advance, before actual participation in consensus, thereby reducing the waiting time when entering the consensus phase. After asynchronously receiving and validating the proposal, each replica immediately sends a preliminary acknowledgment to the primary. This process uses non-blocking transmission and does not require waiting for responses from all replicas. The primary establishes a dynamic threshold based on received acknowledgments and proceeds to the next consensus computation once the number reaches $\mathcal{N} - f$. It also records the status of replicas that fail to respond in time for later compensation. ❷ For decoupling ordering from consensus: After completing local pre-ordering, each replica asynchronously sends its local ordering result along with the merged global candidate sequence (including version metadata) to the primary. The primary does not wait for responses from all replicas; instead, it sets a dynamic confirmation threshold of $\mathcal{N} - f$ and asynchronously accepts ordering results from a majority. Once sufficient preliminary confirmations are received, the primary generates the final order using predefined validation rules, such as version matching, timestamps, or consistent hashing, and broadcasts the result for secondary verification. Upon successful validation, replicas proceed to the next stage of consensus. The system verifies replica-submitted orderings to ensure consistency using digital signatures and hash digests. Lightweight validation reduces computation overhead while maintaining consistency and fairness in ordering. If discrepancies are detected in some replicas' results, the primary triggers local reordering and promptly updates the confirmation status, ensuring the final consensus order remains unaffected by delayed or malicious replicas. ❸ For decoupling computation from communication: Computation and communication tasks are assigned to separate threads or processes. Modules exchange data via shared memory or high-performance message queues using an event-driven model (e.g., asynchronous I/O or non-blocking message queues) for inter-task communication. Tasks such as signature verification and state updates are strictly decoupled from message transmission and vote aggregation, ensuring that the completion of any task can immediately trigger the next stage without waiting for all tasks to finish.

B. *Flexible Decoupling Design*

Specific descriptions are presented in Algorithm 7.

Algorithm 7 Flexible Decoupling Design

```
Input: $R_i \in \mathcal{R}$, $Primary$, MinorMessage
Output: ReProposal
  ▶ Transaction Dissemination
for each $R_i \in \mathcal{R}$ do
    upon receiving committed $\mathsf{TX_k}$ do                    ▷ new $\mathsf{TX_k}$
        if $\mathsf{TXVerify(TX_k)} \wedge \mathsf{TX_k} \notin \mathcal{MEM}$ then
            sign $\mathcal{MEM} := \mathcal{MEM} \cup \{\mathsf{TX_k}\}$            ▷ state update
            update MinorMessage.$Abstract$ ←
                    $\mathsf{Extract}(\mathcal{MEM})$           ▷ generate TX abstracts
            broadcast MinorMessage                       ▷ synchronization
        end if
    end upon
end for
  ▶ Transaction Pre-Validation Call
for each $R_i \in \mathcal{R}$ do
    upon receiving MinorMessage do
        execute Transaction Pre-Validation in Algorithm 3
        update MinorMessage.$Abstract$ ←
                $\mathsf{Extract}(\mathcal{MEM}^*)$              ▷ update TX abstracts
        return $\mathcal{MEM}^*$                                    ▷ mempool update
    end upon
end for
  ▶ Transaction Priority-Queue Call
for each $R_i \in \mathcal{R}$ do
    upon receiving MinorMessage do
        execute Transaction Priority-Queue in Algorithm 4
        return $\mathcal{BUFFER}$                                 ▷ buffer update
    end upon
end for
  ▶ Proposal Generation
for $Primary$ do
    compute TeProposal ←           ▷ generate tentative proposal
            $\mathsf{ProposalGenerate}(\mathcal{BUFFER})$
    broadcast TeProposal
end for
for each $R_i \in \mathcal{R}$ do
    upon receiving TeProposal do                   ▷ validate proposal
        if $\mathsf{QCVerify}$(TeProposal.$QC$) then
            compute ReProposal ←
                    $\mathsf{ReGenerate}$(TeProposal)
        end if                                ▷ generate complete proposal
        broadcast ReProposal
        return ReProposal                           ▷ for next step vote
    end upon
end for
```

Appendix B
Security Analysis

Numerous state-of-the-art consensus protocols have now unanimously chosen to employ threshold signatures to reduce the communication overhead and simplify the verification process. However, it is extremely difficult to guarantee a consensus protocol's adaptive security if it employs a static threshold cryptographic primitive. Therefore, to achieve dynamic defence against adaptive adversaries and high-performance scaling, we adopt an adaptively secure BLS threshold signature, AdaptiveBLS [57], and reimplement it in a large-scale scenario. We follow the mathematical assumptions and proof principles presented in [9], [34], [37], [38], [57]–[59]. Formal proofs of adaptive security for AdaptiveBLS and AdaptiveBFT are as follows.

Reimplementation of AdaptiveBLS [57]. Let $\bar{g}_1, \tilde{g}_1, \hat{g}_1 \in \mathbb{G}_1$ be uniformly random independent generators of $\mathbb{G}_1$. H_0, H_1, and $\mathsf{H}_2 : \{0,1\}^* \to \mathbb{G}_2$ are different cryptographic hash functions modeled as random oracles. Let $\mathsf{sk}_i := (u(i), v(i), w(i))$, $\{\mathsf{pk}_j := \bar{g}_1^{u(j)} \tilde{g}_1^{v(j)} \hat{g}_1^{w(j)}\}_{j \in [n]}$, $\mathsf{pk} := \bar{g}_1^{u(0)} \tilde{g}_1^{v(0)} \hat{g}_1^{w(0)} = \bar{g}_1^{u(0)}$, where $u(\cdot)$, $v(\cdot)$, and $w(\cdot)$ are different uniformly random polynomials of degree t and $v(0) = w(0) = 0$. We introduce Bulletproof, a zero-knowledge proof construction without trustworthy setup, transforming the signature σ_i of i for m into (π_i, σ_i), where π_i is the correctness proof for verification of σ_i provided by Bulletproof. Let the reimplemented AdaptiveBLS be AdaptiveBLS*.

Next, we prove the adaptive security of AdaptiveBLS* with a series of games $\mathsf{SEUF-CMA}_{\Sigma}^{\mathcal{A}}$.

GAME $\mathbf{G_0}$**:** Define the security game $\mathsf{SEUF-CMA}_{\Sigma}^{\mathcal{A}}$, which follows the honest protocol and allows an adaptive adversary $\mathcal{A}$ to access random oracle. Let $\mathcal{A}$ always output the forgery $(\tilde{\sigma}, \tilde{m})$ after querying $\mathsf{H}_0(\tilde{m})$, w.l.o.g., the advantage of $\mathcal{A}$ follows:

$$\mathsf{Adv}_{\mathsf{SEUF-CMA}}^{\mathcal{A},\Sigma}(\lambda) = \Pr\left[\mathbf{G_0} \Rightarrow 1\right] = \varepsilon_\sigma$$

GAME $\mathbf{G_1}$**:** Let $\tilde{m}_r$ be input to r-th random oracle query and $s \xleftarrow{\$} [q_h]$. If $\mathcal{A}$ forges a message $\tilde{m}_r$ with $r \neq s$ or queries over $t - |\mathcal{C}|$ partial signatures for $\tilde{m}_s$, the game aborts. $\mathbf{G_1}$ is identical to $\mathbf{G_0}$, by standard argument, there:

$$\Pr\left[\mathbf{G_1} \Rightarrow 1\right] \geq 1/q_h \cdot \Pr\left[\mathbf{G_0} \Rightarrow 1\right]$$

GAME $\mathbf{G_2}$**:** Let $\xi_{\tilde{g}_1}, \xi_{\hat{g}_1} \xleftarrow{\$} \mathbb{Z}_p$, $\tilde{g}_1 := \bar{g}_1^{\xi_{\tilde{g}_1}}$, and $\hat{g}_1 := \bar{g}_1^{\xi_{\hat{g}_1}}$. $\mathbf{G_2}$ is identical to $\mathbf{G_1}$, by the standard argument, there:

$$\Pr[\mathbf{G_1} \Rightarrow 1] = \Pr[\mathbf{G_2} \Rightarrow 1]$$

GAME $\mathbf{G_3}$**:** Let $\xi := \xi_{\tilde{g}_1} + \omega\xi_{\hat{g}_1}$, where $\omega \xleftarrow{\$} \mathbb{Z}_p$. Others are identical to $\mathbf{G_2}$. Let $\mu_r \xleftarrow{\$} \mathbb{Z}_p$. Only for the r-th random oracle query, the following changes are made to the random oracles:

$$\mathsf{H}_0(\tilde{m}_r) := {g_2}^{\mu_r}$$
$$\mathsf{H}_1(\tilde{m}_r) := {g_2}^{\xi \cdot \mu_r}$$

Lemma 4. $|\Pr[\mathbf{G_2} \Rightarrow 1] - \Pr[\mathbf{G_3} \Rightarrow 1]| \leq \varepsilon_{\mathsf{DDH}} + 1/|\mathbb{Z}_p|$.

Proof. Based on Lemma 2 and Lemma 3 of AdaptiveBLS, distributions D_0 and D_1 are indistinguishable; D_0 and $\mathsf{D}_{1,r}$ are indistinguishable, respectively:

$$\xi \xleftarrow{\$} \mathbb{Z}_p, (\mu_s, \nu_s) \xleftarrow{\$} \mathbb{Z}_p^2;\ \mathsf{D}_0 := g_2, g_2^{\xi}, \{(g_2^{\mu_s}, g_2^{\nu_s})\}_{s \in [q_h]}$$
$$\xi \xleftarrow{\$} \mathbb{Z}_p, \mu_s \xleftarrow{\$} \mathbb{Z}_p; \qquad \mathsf{D}_1 := g_2, g_2^{\xi}, \{(g_2^{\mu_s}, g_2^{\xi \cdot \mu_s})\}_{s \in [q_h]}$$
$$\xi, \mu_s \xleftarrow{\$} \mathbb{Z}_p, \nu_s := \xi \cdot \mu_s; \quad \mathsf{D}_{1,r} := g_2, \{(g_2^{\mu_s}, g_2^{\nu_s})\}_{s \in [q_h]}$$

Thus, samples from distributions D_0 and $\mathsf{D}_{1,s}$ are computationally indistinguishable with:

$$|\Pr[\mathbf{G_2} \Rightarrow 1] - \Pr[\mathbf{G_3} \Rightarrow 1]| \leq \varepsilon_{\mathsf{DDH}} + 1/|\mathbb{Z}_p|$$

□

GAME $\mathbf{G_4}$**:** $\mathbf{G_4}$ is identical to $\mathbf{G_3}$. Simulated Bulletproof provides proof of correctness for partial signatures without revealing information about sk_i. Denote the random oracle query of $\mathcal{A}$ conflicts with the H_2 query as $\mathcal{I}$, then the game abort probability is:

$$\begin{aligned} |\Pr[\mathbf{G_3} \Rightarrow 1] - \Pr[\mathbf{G_4} \Rightarrow 1]| &= |\Pr[\mathbf{G_3} \Rightarrow 1 : \mathcal{I}] \\ &\quad - \Pr[\mathbf{G_4} \Rightarrow 1 : \mathcal{I}]| \cdot \Pr[\mathcal{I}] \\ &\leq \Pr[\mathcal{I}] \\ &\leq (n \cdot q_s \cdot q^*)/|\mathbb{Z}_p|^2 \end{aligned}$$

where q^* is the maximum count of random oracle queries from $\mathcal{A}$ to H_2, and q_s is the maximum count of signature queries (up to n partial signatures per simulation) from $\mathcal{A}$.

GAME $\mathbf{G_5}$**:** Change only the sampling method of the keys for signing. Based on Lemma 6 of AdaptiveBLS, sampling the signature key polynomials for $\mathbf{G_4}$ and $\mathbf{G_5}$, respectively, both of which are random degree t polynomials. Thus, the view of $\mathcal{A}$ is identical in $\mathbf{G_4}$ as in $\mathbf{G_5}$, i.e.,

$$\Pr[\mathbf{G_4} \Rightarrow 1] = \Pr[\mathbf{G_5} \Rightarrow 1]$$

GAME $\mathbf{G_6}$**:** $\mathbf{G_6}$ is identical to $\mathbf{G_5}$. Change only the simulated Bulletproof to actual Bulletproof for partial signatures. Thus, the view of $\mathcal{A}$ is identical in $\mathbf{G_5}$ as in $\mathbf{G_6}$, i.e.,

$$|\Pr[\mathbf{G_5} \Rightarrow 1] - \Pr[\mathbf{G_6} \Rightarrow 1]| \leq (n \cdot q_s \cdot q^*)/|\mathbb{Z}_p|^2$$

Thus, from the series of games above, there:

$$\begin{aligned} |\Pr[\mathbf{G_0} \Rightarrow 1] - \Pr[\mathbf{G_6} \Rightarrow 1]| &\leq (1 - 1/q_h) \cdot \Pr[\mathbf{G_0} \Rightarrow 1] \\ &\quad + \varepsilon_{\mathsf{DDH}} + 1/|\mathbb{Z}_p| \\ &\quad + 2(n \cdot q_s \cdot q^*)/|\mathbb{Z}_p|^2 \\ \Longrightarrow \qquad \Pr[\mathbf{G_6} \Rightarrow 1] &\geq 1/q_h \cdot \varepsilon_\sigma \\ &\quad - \varepsilon_{\mathsf{DDH}} - 1/|\mathbb{Z}_p| \\ &\quad - 2(n \cdot q_s \cdot q^*)/|\mathbb{Z}_p|^2 \end{aligned}$$

Theorem 3 (Adaptive Security of AdaptiveBLS*)**.** *Let* $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_{\mathrm{T}}, \mathbb{F}_p, p)$ *be the public parameters of AdaptiveBLS*. Assuming any PPT adaptive adversary $\mathcal{A}$ that conducts at most q_s signature queries (maximum n partial signatures per session), at most q_h hash queries (also known as random oracle queries) to H_0 and H_1, and at most q^* random oracle queries to H_2 wins the game $\mathsf{SEUF-CMA}_{\Sigma}^{\mathcal{A}}$ with probability:*

$$\varepsilon_\sigma \leq q_h \cdot \left(\varepsilon_{\mathsf{DDH}} + \varepsilon_{\mathsf{CDH}} + 1/|\mathbb{Z}_p| + 2(n \cdot q_s \cdot q^*)/|\mathbb{Z}_p|^2\right)$$

Proof. Based on the formal analysis of $\mathbf{G_0} - \mathbf{G_6}$ above, it is clear that for an adaptive adversary $\mathcal{A}$ that outputs a forgery on message $\tilde{m}_s$ with probability $\varepsilon_\sigma/q_h - \varepsilon_{\mathsf{DDH}} - 1/|\mathbb{Z}_p| - 2(n \cdot q_s \cdot q^*)/|\mathbb{Z}_p|^2$, we can compute the $\mathsf{co-CDH}$ solution efficiently using the forgery on $\tilde{m}_s$. And ε_σ is negligible. Thus, our AdaptiveBLS* is $(T, q_h, q_s, \varepsilon)$-secure against strong existential forgery under adaptive chosen message attacks (SEUF-CMA). This demonstrates that AdaptiveBLS* realizes adaptive security. □

AdaptiveBLS* enhances and ensures the adaptive security of AdaptiveBFT under strong adaptive adversaries, which is proved as follows.

Theorem 4 (Adaptive Security of AdaptiveBFT)**.** *Let* $0 < \varepsilon_{\mathcal{A}} < 1$*. Assume that AdaptiveBFT is a partially synchronous BFT consensus protocol with the threshold signature using AdaptiveBLS***. Then AdaptiveBFT is secure up to* $\mathcal{T} \leq (1-\varepsilon_{\mathcal{A}})\mathcal{N}/2$ *adaptive corruptions, where* $\varepsilon_{\mathcal{A}}^{min} = 1-2t/\mathcal{N}$*.*

Proof. Denote the case where the honest replica $R^* \in \mathcal{H}$ is corrupted by an adaptive adversary $\mathcal{A}$ as $\mathcal{I}^*$. Then, the advantage of $\mathcal{A}$ in the AdaptiveBFT is:

$$\mathsf{Adv}_{\mathsf{BFT}}^{\mathcal{A}}(\lambda) = \Pr\left[\mathcal{I}^*\right] = \varepsilon_{\mathsf{BFT}}$$

Therefore, we have:

$$\varepsilon_{\mathsf{BFT}} \leq q_h \cdot \left(\varepsilon_{\mathsf{DDH}} + \varepsilon_{\mathsf{CDH}} + 1/|\mathbb{Z}_p| + 2(n \cdot q_s \cdot q^*)/|\mathbb{Z}_p|^2\right)$$

Thus, AdaptiveBFT is $(T, q_h, q_s, \varepsilon)$-secure against strong existential forgery under adaptive chosen message attacks (SEUF-CMA). This demonstrates that our AdaptiveBFT realizes adaptive security. $\square$