CAN AI OVERSIGHT BE ZERO-KNOWLEDGE?

Anonymous authors
Paper under double-blind review

ABSTRACT

AI systems increasingly produce outputs from confidential data: a fitness-for-duty assessment from an employee's medical records, or the predicted properties of a drug candidate from its molecular structure, a trade secret. It is important to verify that such outputs are correct without revealing the data on which they are based. A recent line of work studies verification of AI outputs via interactive proofs and debate for oracle-aided computation, where correctness may depend on an oracle such as human judgment, a physical experiment, or an external database such as the web. These works focus on efficient verification, which was shown to be impossible for general oracle-aided computation and so requires additional assumptions on the model. In this work, we focus instead on privacy. To sidestep the impossibility, we allow the verifier to run in time polynomial in the computation, and we ask whether interactive arguments for oracle-aided computation can be zero-knowledge: the verifier learns nothing about the confidential data beyond the correctness of the output. We prove that, in general, they cannot. In the random oracle model, there is no zero-knowledge proofs for all oracle-aided computations, even if both the prover and the verifier are allowed to run much longer than the computation itself. The impossibility extends to debate. On the positive side, we show that if the oracle attaches a cryptographic signature to each of its answers, then every oracle-aided computation can be verified in zero knowledge, assuming only collision-resistant hash functions.

1 INTRODUCTION

AI systems now draft medical assessments (Singhal et al., 2023; Moor et al., 2023; Thirunavukarasu et al., 2023), design candidate molecules (Zhavoronkov et al., 2019; Stokes et al., 2020; Wong et al., 2024), and produce analyses that others rely on (Noy & Zhang, 2023; Brynjolfsson et al., 2025). Although much of the data behind these outputs should remain confidential, it has repeatedly been exposed in practice. DeepSeek left a database with over a million log entries, including chat histories and API secrets, publicly accessible without authentication (Nagli, 2025). Shared ChatGPT conversations and hundreds of thousands of shared Grok conversations became publicly searchable, exposing sensitive personal information (Shanklin, 2025; Martin & Baker-White, 2025). Data brokers reportedly sold searchable chatbot transcripts harvested through browser extensions, including patient information entered by healthcare workers (Claburn, 2026). Researchers have also shown how a crafted email or web page could trick a vulnerable AI assistant into leaking sensitive information (Ravia, 2025; Lakshmanan, 2026).

These incidents motivate stronger privacy protections for data entrusted to AI systems. At the same time, there is growing demand for AI outputs that can be checked efficiently for correctness (Anil et al., 2021; Irving et al., 2018; Brown-Cohen et al., 2024; Kirchner et al., 2024). Even if the leaks described above are prevented, the verification process may itself expose sensitive data. Two notable examples illustrate this.

- *Occupational health.* Recent studies compare AI-generated occupational-health assessments with physicians' judgments (Padovan et al., 2024; Vella et al., 2026). Consider an AI system that drafts a fitness-for-duty assessment from an employee's confidential medical records, which an occupational physician then reviews. The employer, or a licensing body, needs to know whether the employee is "fit to work at heights", but the employee should not have to hand over their medical records. Yet reaching that conclusion requires a physician's judgment about those very records.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

- *Drug discovery.* AI-designed drug candidates now reach clinical trials (Xu et al., 2025; Jayatunga et al., 2024). Before committing to such a candidate, a prospective partner wants evidence that the molecule has the claimed properties. Checking them requires a laboratory *assay*: an experiment, such as a binding or toxicity test, that measures a property of the molecule. Molecular optimization treats the assay as a black-box oracle that the design algorithm queries at a cost (Gao et al., 2022). But running the assay independently requires a physical sample of the molecule, and a sample can be analyzed to recover its structure, which is the trade secret.

Scalable oversight and oracle-aided computation. In both examples, checking the AI's output requires an external judgment or measurement. The same dependence on external resources arises in *scalable oversight*, which asks how a human or a cheaper model can check an AI system's output without carrying out the full task itself (Christiano et al., 2018; Leike et al., 2018; Irving & Marshall, 2025). Following this line of work (Brown-Cohen et al., 2024; Chen et al., 2026), we model an external judgment or resource as a function f, called an *oracle*, which the computation can query in a black-box way.

In this oracle-aided setting, scalable oversight carries out its checks through protocols that build on *interactive proofs* (Goldwasser et al., 1989), in which a *prover* exchanges messages with an efficient *verifier* to establish a claim. The verifier accepts true claims from an honest prover with high probability, while any cheating prover has only a small chance of convincing it to accept a false claim. Here, the AI system acts as the prover: it claims that its output $\mathbb{x}$ is accepted by a public oracle-aided computation D, that is, $D^f(\mathbb{x}) = 1$. Both prover and verifier can query the same oracle. The goal is for the verifier to check this claim reliably while using much less time and fewer oracle queries than running D.

In our occupational-health example, the AI (the prover) must convince the employer or licensing body (the verifier) that the employee is fit to work at heights. The physician is the oracle, answering the medical questions needed to check the assessment. In drug discovery, the AI must convince a prospective business partner that its molecule has the claimed properties. The partner is the verifier, and the oracle answers come from the laboratory assay.

From efficiency to privacy. We ask whether the verifier can check such a claim while the underlying data remains confidential. The prover additionally knows confidential data (the *witness* $\mathbb{w}$) that the verifier does not receive, and the claim becomes "there exists $\mathbb{w}$ such that $D^f(\mathbb{x}, \mathbb{w}) = 1$". In our examples, $\mathbb{w}$ is the medical records or the molecular structure, and $\mathbb{x}$ is the published conclusion, together with information identifying the confidential object it concerns. The employer or business partner should be convinced that a valid witness supports the claim, without learning anything about the confidential data. This privacy requirement can be formalized using *zero knowledge* (Goldwasser et al., 1989): the verifier can check the correctness of the claim without learning anything about the hidden data beyond what the claim itself reveals.

Our model: zero-knowledge oversight. Ideally, checking an AI's output would be both efficient and zero-knowledge. However, Barbara et al. (2025) establish limitations on efficient verification for oracle-aided computations, even without a privacy requirement. But an employer or a business partner would be willing to spend more time checking a conclusion if that is what protecting the underlying data requires. We therefore relax the efficiency requirement and ask:

Can every oracle-aided computation be verified in zero knowledge, even if we do not require efficient verification?

1.1 Our results

We answer this question in the negative: not every oracle-aided computation can be verified in zero knowledge, even if we do not require efficient verification. To work around this impossibility, we construct a protocol for efficient zero-knowledge verification under standard cryptographic assumptions by modifying the oracle to provide digital signatures on its answers. Our construction is therefore a compromise: in some applications, the oracle cannot provide these signatures itself and requires a trusted third party to do so, as we discuss below.

The impossibility result. Our impossibility result holds in the *random oracle model*, where the oracle is a uniformly random function. This is the standard model for relativized arguments (Chiesa

& Tromer, 2010; Chen et al., 2022), and the model in which the efficiency limitation of Barbara et al. (2025) is proved. We prove it even after weakening both the correctness and privacy requirements: (i) we only require the verifier to reject false claims from polynomial-time cheating provers, as in *relativized arguments*; (ii) we weaken privacy to *witness hiding*: the verifier may learn partial information about the confidential data, but should not be able to recover it in full or find alternative data that would pass the check for the same claim.

Theorem 1 (informal version of Theorem 3.3)**.** *There exists an oracle-aided computation in the random oracle model that has no witness-hiding relativized argument, even when efficient verification is not required.*

Since zero knowledge implies witness hiding, this rules out zero-knowledge verification as well.

Our lower bound concerns single-prover verification, the setting studied by Chen et al. (2026) for oracle-aided scalable oversight. Recall that Barbara et al. (2025) rules out efficient single-prover verification for general oracle-aided computations. Chen et al. (2026) circumvents this limitation by assuming the computation is *robust*: its output tolerates a small fraction of incorrect oracle answers. This allows the verifier to check the computation using only a few oracle queries. But using similar assumption in zero knowledge would in addition require the verifier's queries and the answers it receives to reveal nothing about the hidden witness. This need not hold in general for computations we care about. For example, a query asking whether a molecule with a specified structure binds to a target reveals that entire structure to the verifier.

Another canonical model for scalable oversight is *debate*, where two provers compete to convince a verifier and correctness relies on the strong assumption that one prover acts honestly (Irving et al., 2018; Brown-Cohen et al., 2024). This additional assumption allows debate to circumvent the efficiency limitation of Barbara et al. (2025). We can define privacy for debate in two ways, depending on which provers receive the secret witness. However, neither definition allows zero-knowledge verification in general. If both provers receive it, the malicious prover would reveal it to the verifier. If only the honest prover receives it, our impossibility still applies. See Section 2.2 for details.

We therefore need a different approach to obtain zero knowledge for general oracle-aided computations.

Construction with signed oracles. Chiesa & Tromer (2010) introduced the *signed random oracle model*, in which the random oracle returns each answer together with a digital signature on the query and its answer, and constructed relativized arguments with efficient verification in this model. We give a general oracle the same interface. A digital signature scheme lets the holder of a secret signing key sk produce a signature σ on any string m. Anyone holding the matching public verification key vk can then check that σ is a valid signature on m, and without sk nobody can produce a valid signature on a new string. An oracle f answers a query q with an answer a:

$$f(q) = a \ .$$

Its signed version $\widehat{f}$ returns the same answer together with a signature on the query and the answer:

$$\widehat{f}(q) = (a, \sigma) \text{ where } \sigma \leftarrow \mathsf{Sign}\big(\mathsf{sk}, (q, a)\big) \ .$$

The signature certifies that a is the oracle's answer to q, and anyone holding vk can check this without querying f.

With this interface, we obtain zero-knowledge arguments with efficient verification for all oracle-aided computations.

Theorem 2 (informal version of Theorem 3.5)**.** *Assume the existence of collision-resistant hash functions. For every oracle f and every oracle-aided computation, there exists a zero-knowledge single-prover argument with efficient verification using the signed oracle $\widehat{f}$.*

This is the compromise mentioned at the start of this section. The signed oracle $\widehat{f}$ exists only if some party holds the signing key, and that party sees every query. It must be trusted to sign only genuine answers and to keep the queries private. Whether such a party is available depends on the application, but it is a plausible assumption in many natural cases. When the oracle is a digital resource, such as a database or a model, the signer can be trusted hardware that evaluates the oracle and signs its answers, as studied in cryptography (Katz, 2007; Pass et al., 2017), in secure systems and databases (McKeen

et al., 2013; Costan & Devadas, 2016; Bajaj & Sion, 2014; Priebe et al., 2018), and in machine learning (Tramèr & Boneh, 2019; Mo et al., 2021; Rowstron, 2026). In the occupational-health example, the physician can act as the signer and sign the judgments used in the assessment. In the drug-discovery example, the assay itself cannot sign anything: a laboratory must run the assay and sign the results, and it must be trusted with the sample and the measurements.

1.2 Related work

Private verification of AI systems. Cryptographic verification of machine-learning computations includes zero-knowledge proofs of neural-network inference, from convolutional networks (Liu et al., 2021) to large language models (Sun et al., 2024), proof systems optimized for machine-learning operations (Chen et al., 2024), and zero-knowledge proofs of training (Garg et al., 2023; Abbaszadeh et al., 2024). Other works establish that privacy and verifiability are compatible when the model is audited for a certain criteria if the audit is specified as an ordinary computation: ZkAudit verifies computations over private models and training data (Waiwitlikhit et al., 2024) and OATH certifies fairness of decisions while protecting model and user data (Franzese et al., 2025). On the other hand, our question concerns checks whose validity also depends on external oracle answers.

Scalable oversight. Scalable oversight concerns reliable supervision or evaluation of an AI model, even when the task/model exceeds a human judge's capabilities. Works such as (Christiano et al., 2018; Leike et al., 2018; Burns et al., 2024) focus on the scalable supervision in the training. On the other hand, Bowman et al. (2022) propose an empirical methodology for measuring progress on scalable oversight. Irving et al. (2018) propose debate as an alternative, using competing provers to help a weaker verifier. Brown-Cohen et al. (2024) formalize doubly-efficient debate for oracle-aided tasks, building on doubly-efficient interactive proofs (Goldwasser et al., 2015). Buhl et al. (2025) examine the relevance of debate model to the alignment safety and identify the assumptions needed. Irving & Marshall (2025) explain the role of relativization when verification depends on human judgments or other external resources. Barnes (2020) identifies the obfuscated-arguments problem for debate and suggests that "small" arguments are needed. Later work (Brown-Cohen et al., 2025) addresses obfuscation by prover-estimator debate. Chen et al. (2026) show that we can consider the simpler model with a single prover if the oracle-aided computation is robust or the oracle is low-degree. These works principally address correctness and the cost of oversight, rather than simulation-based privacy for a confidential witness. We keep their oracle-aided formulation but ask whether the witness can remain hidden even when the verifier is allowed a polynomial overhead.

Relativized arguments. The study of relativized arguments is also motivated by proofs that verify other proofs, as in incrementally verifiable computation and proof-carrying data (Valiant, 2008; Chiesa & Tromer, 2010; Bitansky et al., 2013). The issue is that the computation being proved may invoke the same oracle as the proof system. Positive constructions exploit signed responses (Chiesa & Tromer, 2010; Maller et al., 2025), low-degree encodings (Chen et al., 2022), or arithmetized oracle access (Chen et al., 2023). Chiesa et al. (2024) analyze the security of recursive proof composition in oracle models. Oracle-relative formulations also arise in the study of white-box security for the Fiat–Shamir transformation (Arnon & Yogev, 2025).

On the negative side, Chiesa & Liu (2020) establish limitations of probabilistically checkable proofs and interactive oracle proofs in relativized worlds. Most directly, Barbara et al. (2025) rule out succinct relativized arguments in the random oracle model (Bellare & Rogaway, 1993), even with interaction and computational soundness. Our lower bound is complementary: it imposes no succinctness requirement and rules out even witness hiding (Feige & Shamir, 1990), a requirement weaker than zero knowledge (Goldwasser et al., 1989).

2 Technical overview

We explain the main ideas behind our results. Section 2.1 explains the proof of the lower bound (Theorem 1), Section 2.2 explains why the lower bound extends to debate, and Section 2.3 explains the construction with signed oracles (Theorem 2).

2.1 THE LOWER BOUND

Recall that our goal is to construct a *relativized interactive argument* for an oracle-aided computation D with access to an arbitrary oracle f. The prover receives $(\mathbb{x}, \mathbb{w})$, while the verifier receives only $\mathbb{x}$; both can query f. The prover must convince the verifier that there exists a private witness $\mathbb{w}$ such that $D^f(\mathbb{x}, \mathbb{w}) = 1$. Correctness requires the verifier to accept true claims from an honest prover (*completeness*) and reject false claims from computationally bounded cheating provers (*soundness*), with high probability. Zero knowledge requires the interaction to reveal nothing about $\mathbb{w}$. Our lower bound rules out even *witness hiding*, the weaker requirement of Section 1.1: the verifier may learn partial information about a witness, as long as what it learns is not enough to recover an entire valid witness. Since zero knowledge implies witness hiding, ruling out witness hiding rules out zero knowledge as well, and we speak of witness hiding from here on.

Our impossibility result comes from a conflict between correctness and witness hiding. We choose f to be a *random oracle*: a function sampled uniformly at random and then kept fixed. A protocol for arbitrary oracles would have to work for this family too. We construct a computation $C_\star$ for which any correct protocol forces the verifier to learn a valid witness with constant probability. A witness cannot be found from the public input and oracle access alone, so a protocol that reveals one to the verifier gives it something it could never have computed itself. This is exactly what witness hiding forbids.

A hard computation. We use the computation underlying the relation of Barbara et al. (2025). Fix the public input $\mathbb{x} = 0^n$ and let candidate witnesses be m-bit strings. The computation accepts exactly when

$$C_\star^f(0^n, \mathbb{w}) = 1 \quad \Longleftrightarrow \quad f(\mathbb{w} \| i)_1 = 0 \quad \text{for every } i \in \{1, \ldots, m\}. \tag{1}$$

Here $\|$ denotes concatenation, i is written as a binary string of fixed length, and the subscript 1 denotes the first bit of the oracle's answer.

Think of these bits as a table with 2^m rows and m columns (Figure 1). Each row is labeled by a candidate witness $\mathbb{w}$, and column i contains $f(\mathbb{w}\|i)_1$. The claim is that the table contains an all-zero row, and the witness is the label of such a row. Given the witness, $C_\star$ checks the claim by reading the m bits in that row. Without the witness, finding an all-zero row is hard: the rows are independent, each is all zero with probability 2^{-m}, and polynomially many oracle queries have only an exponentially small chance of finding one. Finally, the query $\mathbb{w}\|i$ contains the entire row label, so a verifier that queries even one entry of a valid row has written down a valid witness.

The table is a simplified version of the drug-discovery example. A row label is the structure of a molecule, a column is a test, and a zero means that the molecule passed that test. The claim is that some molecule passes every test, and a witness is such a molecule. Asking for a test result means submitting the structure, so a verifier cannot ask about a molecule without revealing it. Our lower bound also assumes that the test outcomes are independent random bits, which is what the random oracle provides.

The verifier must read the witness row. Suppose, towards a contradiction, that $(\mathcal{P}, \mathcal{V})$ is a witness-hiding relativized argument for $C_\star$. Consider a cheating prover $\widetilde{\mathcal{P}}$ that chooses a random row label

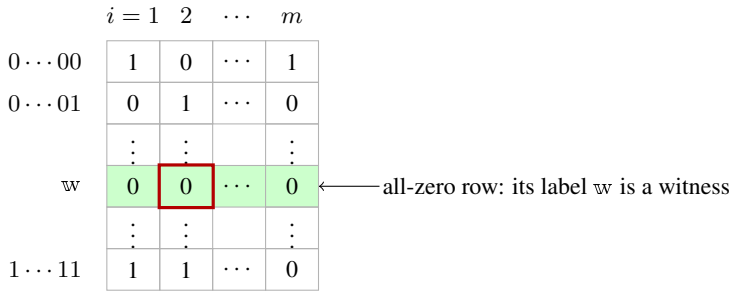


Figure 1: The table of bits read by the hard computation $C_\star$: 2^m rows, one per candidate witness, and m columns. A witness is the label of an all-zero row (shaded); the boxed entry is what one query reads.

$\mathbb{w}$ and runs $\mathcal{P}$ on $(0^n, \mathbb{w})$, but answers its oracle queries using a modified oracle f' (Figure 2). This oracle agrees with f except that the first bits of all answers in row $\mathbb{w}$ are set to zero. $\widetilde{\mathcal{P}}$ can supply these altered answers to its internal copy of $\mathcal{P}$; the verifier $\mathcal{V}$ still queries the real oracle f.

Now suppose the real table has no all-zero row, which happens with constant probability. Then $C_\star^f(0^n, \mathbb{w}') = 0$ for every $\mathbb{w}'$, while $C_\star^{f'}(0^n, \mathbb{w}) = 1$. Write $\mathcal{V}^f \in \{0, 1\}$ for the decision of $\mathcal{V}$ when its oracle is f, and consider two interactions in which the prover sends exactly the same messages.

- *The honest interaction* between $\mathcal{P}$ and $\mathcal{V}$, both with oracle f'. By completeness, $\mathcal{V}^{f'} = 1$ with high probability, since $C_\star^{f'}(0^n, \mathbb{w}) = 1$.
- *The cheating interaction* between $\widetilde{\mathcal{P}}$ and $\mathcal{V}$, where $\mathcal{V}$ queries f. By soundness against $\widetilde{\mathcal{P}}$, $\mathcal{V}^f = 0$ with high probability, since $C_\star^f(0^n, \mathbb{w}') = 0$ for every $\mathbb{w}'$.

The two oracles agree outside row $\mathbb{w}$. As long as $\mathcal{V}$ does not query row $\mathbb{w}$, it receives the same oracle answers and the same prover messages in both interactions, and therefore reaches the same decision. Since the two decisions must differ, $\mathcal{V}$ queries row $\mathbb{w}$ with high probability in the honest interaction with f'. Making that query requires $\mathcal{V}$ to know $\mathbb{w}$. With some additional care, the same conclusion holds when the oracle is a random oracle on a true claim rather than the modified oracle f'. So the honest verifier learns a witness that it could not have found on its own, which violates witness hiding, and hence zero knowledge as well. Nothing in the argument required $\mathcal{V}$ to be efficient, which is why the theorem needs no efficiency requirement (Theorem 3.3).

2.2 Debate

We now show that the lower bound extends to debate. This contrasts with efficiency: a second prover lets debate achieve efficient verification where a single prover cannot (Brown-Cohen et al., 2024), but it does not make privacy possible. Recall that a *debate* involves a prover $\mathcal{P}_1$, an opponent $\mathcal{P}_2$, and a verifier $\mathcal{J}$ (Irving et al., 2018; Brown-Cohen et al., 2024). All three know the public input $\mathbb{x}$ and can query the oracle f. The prover $\mathcal{P}_1$ tries to establish that there exists a witness $\mathbb{w}$ such that $D^f(\mathbb{x}, \mathbb{w}) = 1$, while the opponent $\mathcal{P}_2$ challenges this claim. Completeness requires that, if the claim is true, the honest prover $\mathcal{P}_1$ convinces $\mathcal{J}$ to accept regardless of the opponent's strategy. Soundness requires that, if the claim is false, the honest opponent $\mathcal{P}_2$ makes $\mathcal{J}$ reject regardless of the cheating prover's strategy.

There are two natural ways to define privacy for debate, depending on which provers receive the secret witness. One option is to give $\mathbb{w}$ to both $\mathcal{P}_1$ and $\mathcal{P}_2$ and require that it remain hidden from $\mathcal{J}$. This fails in general because a malicious $\mathcal{P}_2$ can simply send $\mathbb{w}$ to $\mathcal{J}$. The other option is to give $\mathbb{w}$ only to $\mathcal{P}_1$, while $\mathcal{P}_2$ and $\mathcal{J}$ receive only $\mathbb{x}$ and oracle access. In this case, privacy must hold against $\mathcal{P}_2$ and $\mathcal{J}$ together, since they may cooperate. Together, they must not be able to recover $\mathbb{w}$.

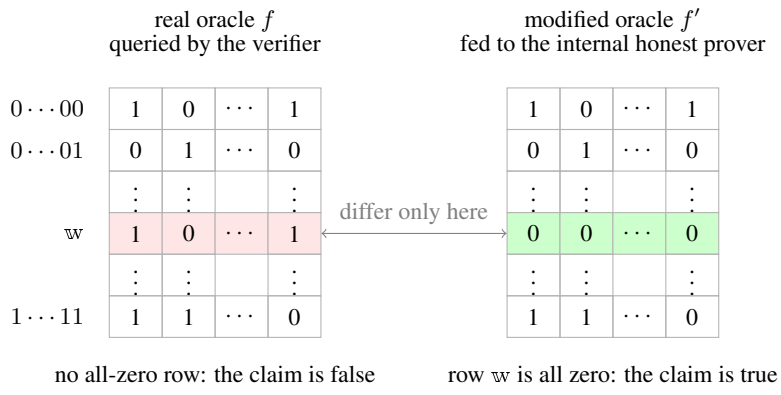


Figure 2: The two worlds in the proof. Left: the real oracle f, which the verifier queries; no row is all zero, so the claim is false. Right: the oracle f' that the cheating prover feeds to its internal copy of the honest prover; row $\mathbb{w}$ is overwritten with zeros, so the claim is true and $\mathbb{w}$ is a witness. The tables agree outside row $\mathbb{w}$, so a verifier that never reads that row receives the same oracle answers, and the same prover messages, in both worlds.

For this second definition, our lower bound applies by a reduction to the single-prover case. Let $\mathcal{V}$ be the verifier that runs $\mathcal{P}_2$ and $\mathcal{J}$ internally, exchanges messages with $\mathcal{P}_1$ on their behalf, and outputs the decision of $\mathcal{J}$. Then $(\mathcal{P}_1, \mathcal{V})$ is a single-prover argument for the same claim:

- it is complete, because the honest $\mathcal{P}_1$ convinces $\mathcal{J}$ against every opponent, in particular against $\mathcal{P}_2$;
- it is sound, because the soundness of the debate is exactly the guarantee that $\mathcal{P}_2$ defeats every cheating $\widetilde{\mathcal{P}}_1$;
- it is witness hiding if the debate is, because $\mathcal{V}$ knows exactly what $\mathcal{P}_2$ and $\mathcal{J}$ know together.

Hence a witness-hiding debate for the hard computation $C_\star$ would give a witness-hiding argument for it, which Theorem 1 rules out. Corollary 3.4 states the result, and Section E gives the reduction (Lemma E.6).

2.3 Construction with signed oracles

We now show how a signed oracle makes efficient zero-knowledge verification possible for every oracle-aided computation (Theorem 2). Recall from Section 2.1 that the lower bound rests on one fact: the only way for the verifier to check that an oracle answer is genuine is to make the query, and for some computations the query requires knowing the witness. A signed oracle $\widehat{f}$ (Section 1.1) removes this obstacle. Every answer comes with a signature on the query and the answer, and anyone holding vk can check the signature without querying the oracle, so certifying an answer no longer requires querying the oracle itself.

Here is a first attempt. The prover runs D on $(\mathbb{x}, \mathbb{w})$ with the signed oracle and sends the verifier the entire record of the execution: every query it made, the answer, and the signature. The verifier checks that the record is an accepting execution of D and that every signature in it is valid under vk, without querying the oracle. This is sound, but it is not zero knowledge: the record contains the prover's queries, which contain the confidential data. So the prover must keep $\mathbb{w}$ and the record to itself, and instead prove to the verifier that it holds a witness and a record such that (i) the record is an accepting execution of D on $(\mathbb{x}, \mathbb{w})$ and (ii) every signature in the record is valid under vk. Given $\mathbb{w}$ and the record, checking (i) and (ii) is an ordinary computation that never calls f.

Specifically, the prover proves that for $(\mathbb{x}, \mathsf{vk})$ there exist $\mathbb{w}$ and $((q_i, a_i, \sigma_i))_i$ such that D on $(\mathbb{x}, \mathbb{w})$ makes the queries $q_1, q_2, \ldots$, receives the answers $a_1, a_2, \ldots$, and accepts, and $\mathsf{Verify}(\mathsf{vk}, (q_i, a_i), \sigma_i) = 1$ for every i. This is an NP statement that does not depend on any oracle: given $\mathbb{w}$ and $((q_i, a_i, \sigma_i))_i$, it can be checked in polynomial time. Thus we can use standard techniques from cryptography to prove it, namely a *zero-knowledge argument of knowledge*. Such an argument lets a prover convince a verifier that it knows a valid witness for an NP statement while revealing nothing about that witness beyond the truth of the statement. Arguments of this kind with efficient verification exist assuming collision-resistant hash functions (Kilian, 1992; Barak & Goldreich, 2008).

We apply this argument to the statement above, with public input $(\mathbb{x}, \mathsf{vk})$ and private witness $(\mathbb{w}, ((q_i, a_i, \sigma_i))_i)$. The resulting protocol has the three properties we want:

- it is sound, because a fabricated oracle answer in the record would come with a forged signature, which the knowledge requirement rules out;
- it is zero knowledge, because the argument hides the original witness, the queries, and the answers;
- it has efficient verification, because the verifier never queries the underlying oracle f, and checks a succinct argument.

Combining a signed oracle with an ordinary zero-knowledge argument of knowledge therefore gives efficient zero-knowledge verification of the original oracle-aided computation (Theorem 3.5).

3 Formal statements

We now state our results formally. Definitions are given in full in Section A, and proofs in Sections D to F. All results concern the following objects; λ denotes the security parameter and n the instance size.

Definition 3.1 (Oracle, relativized relation)**.** *An* oracle *is a collection $\mathcal{O} = \{\mathcal{O}_{\lambda,n}\}_{\lambda,n\in\mathbb{N}}$, where each $\mathcal{O}_{\lambda,n}$ is a distribution over functions $X_{\lambda,n} \to Y_{\lambda,n}$ for finite sets $X_{\lambda,n}$ and $Y_{\lambda,n}$; we write $f \in \mathcal{O}$ for a function in the support of some $\mathcal{O}_{\lambda,n}$. The* random oracle $\mathcal{U}$ *over $(X_{\lambda,n}, Y_{\lambda,n})_{\lambda,n\in\mathbb{N}}$ is the oracle in which each $\mathcal{U}_{\lambda,n}$ is the uniform distribution over such functions. A* relativized relation *for $\mathcal{O}$ is a collection $R_{\mathcal{O}} = \{R_f\}_{\lambda,n\in\mathbb{N},\, f\in\mathcal{O}_{\lambda,n}}$ of relations, and its language is $L(R_f) \coloneqq \{\mathbb{x} : \exists\, \mathbb{w}$ with $(\mathbb{x},\mathbb{w}) \in R_f\}$ (Definitions A.2 and A.4).*

Definition 3.2 (Relativized interactive argument)**.** *A* relativized interactive argument *for $R_{\mathcal{O}}$ is a pair $(\mathcal{P},\mathcal{V})$ of interactive oracle algorithms: $\mathcal{P}$ receives $(\mathbb{x},\mathbb{w})$, $\mathcal{V}$ receives $\mathbb{x}$, both query the same function f, and $\mathcal{V}$ outputs a decision $\mathsf{d} \in \{0,1\}$, written $\mathsf{d} \leftarrow \langle \mathcal{P}^f(1^\lambda,\mathbb{x},\mathbb{w}), \mathcal{V}^f(1^\lambda,\mathbb{x})\rangle$. It has* completeness error *α if for every λ, n, every $\mathbb{x} \in \{0,1\}^n$, and every oracle-aided algorithm $\mathcal{A}$,*

$$\Pr\left[\, \mathsf{d} = 0 \,\wedge\, (\mathbb{x},\mathbb{w}) \in R_f \;\middle|\; f \leftarrow \mathcal{O},\ \mathbb{w} \leftarrow \mathcal{A}^f(\mathbb{x}),\ \mathsf{d} \leftarrow \langle \mathcal{P}^f(1^\lambda,\mathbb{x},\mathbb{w}), \mathcal{V}^f(1^\lambda,\mathbb{x})\rangle \,\right] \le \alpha(\lambda,n),$$

and soundness error *β if for every λ, n, every $\mathbb{x} \in \{0,1\}^n$, and every q-query t-time cheating prover $\widetilde{\mathcal{P}}$,*

$$\Pr\left[\, \mathsf{d} = 1 \,\wedge\, \mathbb{x} \notin L(R_f) \;\middle|\; f \leftarrow \mathcal{O},\ \mathsf{d} \leftarrow \langle \widetilde{\mathcal{P}}^f(\mathbb{x}), \mathcal{V}^f(1^\lambda,\mathbb{x})\rangle \,\right] \le \beta(\lambda,n,\mathsf{q},\mathsf{t}) \ .$$

Its efficiency is measured by the numbers of oracle queries and the running times of the verifier and of the honest prover (Definition A.16).

3.1 The witness-hiding lower bound

For the computation $C_\star$ from Section 2.1, we allow both the prover and the verifier to run in time polynomial in the cost of checking a witness. The verifier need not be faster than the computation it checks. Even with this allowance, correctness and witness hiding cannot both be guaranteed.

Let λ be the security parameter, n the instance size, and $t(n) \ge n$ the witness length, so that checking a witness for $C_\star$ costs $t(n)$ oracle queries; this is Equation 1 with $m = t(n)$, where the random oracle $\mathcal{U}$ maps inputs of length $t(n) + O(\log t(n))$ to $\{0,1\}^\lambda$. Write $R_{\mathcal{U}}$ for the resulting relation, whose instances are 0^n and whose witnesses are the strings $\mathbb{w} \in \{0,1\}^{t(n)}$ with $C^u_\star(0^n,\mathbb{w}) = 1$ (Definition B.4). Besides the completeness and soundness errors of Definition 3.2, a relativized argument $(\mathcal{P},\mathcal{V})$ for it has a *witness hiding error* against aq-query at-time adversaries: the probability that an adversary given the honest verifier's randomness and the transcript of an honest interaction outputs a valid witness (Definition A.17). Since zero knowledge implies witness hiding, the results below rule out zero knowledge as well.

Theorem 3.3 (Witness-hiding lower bound)**.** *No relativized argument for $R_{\mathcal{U}}$ in which the verifier and the honest prover make $\mathsf{poly}(\lambda,t(n))$ queries and run in time $\mathsf{poly}(\lambda,t(n))$ has all of the following: completeness error at most $1/10$, soundness error at most $1/10$ against $\mathsf{poly}(\lambda,t(n))$-query $\mathsf{poly}(\lambda,t(n))$-time cheating provers, and witness hiding error at most $1/10$ against $\mathsf{poly}(\lambda,t(n))$-query $\mathsf{poly}(\lambda,t(n))$-time adversaries.*

Theorem D.1 states the theorem with all parameters explicit. It follows from a quantitative tradeoff. Fix any relativized argument for $R_{\mathcal{U}}$ whose honest prover and verifier run in time $\mathsf{poly}(\lambda,t(n))$, and write $\varepsilon_{\mathrm{complete}}$ for its completeness error, $\varepsilon_{\mathrm{sound}}$ for its soundness error against cheating provers with the resources of the honest prover, and $\varepsilon_{\mathrm{hide}}$ for its witness hiding error against adversaries with the resources of the verifier, in both cases up to a factor of $t(n)$. Then (Lemma D.2)

$$\varepsilon_{\mathrm{complete}} + \varepsilon_{\mathrm{sound}} + \varepsilon_{\mathrm{hide}} \;\ge\; \frac{1}{e} - o(1) \ .$$

The right-hand side is the probability that the claim is false, so the three errors compete for a fixed budget. Making the verifier more reliable does not remove the privacy problem: as the completeness and soundness errors decrease, the lower bound on the probability of recovering the witness increases. The constant $1/10$ in Theorem 3.3 is therefore not essential.

3.2 The extension to debate

The same lower bound applies to the private-witness debate model of Section 2.2. Here the opponent receives no witness, and privacy must hold against the opponent and the verifier together. Formally, a relativized debate $(\mathcal{P}_1,\mathcal{P}_2,\mathcal{J})$ for $R_{\mathcal{U}}$ gives $(\mathbb{x},\mathbb{w})$ to the prover and $\mathbb{x}$ to the opponent and the

verifier, all three query the oracle, and the verifier decides after exchanging messages with both provers (Section E). Its completeness error is measured against every q-query t-time opponent, its soundness error against every q-query t-time cheating prover facing the honest opponent, and its witness hiding error against adversaries that receive the instance, the randomness of the opponent and the verifier, and the transcript (Definition E.3).

Corollary 3.4 (Lower bound for debate)**.** *No relativized debate for $R_{\mathcal{U}}$ in which the verifier, the honest prover, and the honest opponent make $\mathsf{poly}(\lambda, t(n))$ queries and run in time $\mathsf{poly}(\lambda, t(n))$ has all of the following: completeness error at most $1/10$ against $\mathsf{poly}(\lambda, t(n))$-query $\mathsf{poly}(\lambda, t(n))$-time opponents, soundness error at most $1/10$ against $\mathsf{poly}(\lambda, t(n))$-query $\mathsf{poly}(\lambda, t(n))$-time cheating provers, and witness hiding error at most $1/10$ against $\mathsf{poly}(\lambda, t(n))$-query $\mathsf{poly}(\lambda, t(n))$-time adversaries.*

The corollary follows from Theorem 3.3 through the reduction of Section 2.2, which Lemma E.6 makes precise; see Corollary E.7.

3.3 Zero knowledge with signed oracles

With a signed oracle, the guarantee is stronger than witness hiding. Even a verifier that deviates from the protocol learns nothing beyond the public claim: its entire view can be simulated without the witness, using the same fixed oracle. Formally, an argument has *malicious-verifier zero-knowledge error z* if a simulator with black-box access to the malicious verifier, which it may run and rewind, produces a distribution within statistical distance z of the verifier's view, which consists of its randomness, the messages it receives, and its oracle queries and answers (Definition A.19). The signed oracle $\widehat{\mathcal{O}}$ is the one of Section 1.1; it also answers the query "key" with the verification key (Definition F.3). The computations covered are all oracle circuits: an instance is the description of an oracle circuit $\mathcal{C}$ together with an input $\mathbb{x}'$, and a witness is a string $\mathbb{w}$ with $\mathcal{C}^f(\mathbb{x}'\|\mathbb{w}) = 1$ (Definition F.2).

Theorem 3.5 (Zero knowledge with a signed oracle)**.** *Let $\mathcal{O}$ be an oracle with one-bit answers, and assume a collision-resistant hash family that is secure against adversaries with oracle access to $\mathcal{O}$. Then oracle-circuit satisfiability relative to $\mathcal{O}$ has a relativized argument in the model of the signed oracle $\widehat{\mathcal{O}}$ with the following guarantees.*

- Completeness *is perfect.*
- Soundness *holds computationally: a cheating prover running in time $\mathsf{poly}(\lambda, n)$ convinces the verifier of a false claim with probability at most $1/2 + \mathrm{negl}(\lambda)$.*
- Zero knowledge *holds statistically against malicious verifiers, with error $2^{-\lambda}$ and a simulator running in time polynomial in the running time of the verifier.*
- Efficiency*: the communication and the running time of the verifier are $\mathsf{poly}(\lambda, n)$, and the honest verifier queries the oracle only once, to retrieve the verification key.*

Here n is the length of the instance, which includes the description of the circuit; the verifier's cost is polynomial in the description of the computation, and the verifier never evaluates the oracle, which in our applications is the expensive step. Standard sequential repetition reduces the soundness error to negligible while preserving the other guarantees. The argument combines the signed oracle with a zero-knowledge argument of knowledge relative to an oracle (Theorem F.6); see Theorem F.4.

AI use statement

We used generative AI tools to help write some parts of the paper. The main text (Sections 1 to 3) was written by the authors, with AI assistance to improve the language and to find grammatical mistakes. The lower bound and its proof (Sections B to D) are entirely written by the authors, apart from typo and spelling checks. For the extension to debate (Section E) and the construction with signed oracles (Section F), the ideas originated from the authors; the first drafts of the proofs were produced by human-guided AI generation, and the current versions were modified and checked by the authors to make sure that they match the intended arguments. We have not used generative AI tools for any other purposes with required disclosure. Additionally, we used generative AI tools to search for and summarize relevant literature, to prepare the figures, and to obtain feedback on drafts. We have reviewed all AI-assisted work: every reference identified by an LLM was checked manually by at least one of the authors, and all AI-assisted text was revised by the authors. We take responsibility for

the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Kasra Abbaszadeh, Christodoulos Pappas, Jonathan Katz, and Dimitrios Papadopoulos. Zero-knowledge proofs of training for deep neural networks. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, CCS '24, pp. 4316–4330, 2024. doi: 10.1145/3658644.3670316.

Cem Anil, Guodong Zhang, Yuhuai Wu, and Roger Grosse. Learning to give checkable answers with prover–verifier games, 2021. arXiv:2108.12099.

Gal Arnon and Eylon Yogev. Towards a white-box secure fiat-shamir transformation. In *Proceedings of the 45th Annual International Cryptology Conference*, CRYPTO '25, pp. 27–56, 2025.

László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, STOC '91, pp. 21–32, 1991.

Sumeet Bajaj and Radu Sion. TrustedDB: A trusted hardware-based database with privacy and data confidentiality. *IEEE Transactions on Knowledge and Data Engineering*, 26(3):752–765, 2014. doi: 10.1109/TKDE.2013.38.

Boaz Barak and Oded Goldreich. Universal arguments and their applications. *SIAM Journal on Computing*, 38(5):1661–1694, 2008. Preliminary version appeared in CCC '02.

Annalisa Barbara, Alessandro Chiesa, and Ziyi Guan. Relativized succinct arguments in the ROM do not exist. In *Proceedings of the 23rd Theory of Cryptography Conference*, TCC '25, pp. 400–416, 2025.

Beth Barnes. Debate update: Obfuscated arguments problem. AI Alignment Forum post, 2020. URL `https://www.alignmentforum.org/posts/PJLABqQ962hZEqhdB/debate-update-obfuscated-arguments-problem`.

Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, CCS '93, pp. 62–73, 1993.

Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. Recursive composition and bootstrapping for SNARKs and proof-carrying data. In *Proceedings of the 45th ACM Symposium on the Theory of Computing*, STOC '13, pp. 111–120, 2013.

Samuel R. Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamilė Lukošiūtė, Amanda Askell, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Christopher Olah, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Jackson Kernion, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Liane Lovitt, Nelson Elhage, Nicholas Schiefer, Nicholas Joseph, Noemí Mercado, Nova DasSarma, Robin Larson, Sam McCandlish, Sandipan Kundu, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Ben Mann, and Jared Kaplan. Measuring progress on scalable oversight for large language models, 2022. arXiv:2211.03540.

Jonah Brown-Cohen, Geoffrey Irving, and Georgios Piliouras. Scalable AI safety via doubly-efficient debate. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4585–4602. PMLR, 2024.

Jonah Brown-Cohen, Geoffrey Irving, Georgios Piliouras, Lijie Chen, Jiawei Li, and Zhiyang Xun. Avoiding obfuscation with prover-estimator debate, 2025. arXiv preprint arXiv:2506.13609.

Erik Brynjolfsson, Danielle Li, and Lindsey Raymond. Generative AI at work. *The Quarterly Journal of Economics*, 140(2):889–942, 2025. doi: 10.1093/qje/qjae044.

Marie Davidsen Buhl, Jacob Pfau, Benjamin Hilton, and Geoffrey Irving. An alignment safety case sketch based on debate, 2025. arXiv preprint arXiv:2505.03989.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeffrey Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML '24*, pp. 4971–5012, 2024.

Bing-Jyue Chen, Suppakit Waiwitlikhit, Ion Stoica, and Daniel Kang. ZKML: An optimizing system for ML inference in zero-knowledge proofs. In *Proceedings of the 19th European Conference on Computer Systems*, EuroSys '24, pp. 560–574, 2024. doi: 10.1145/3627703.3650088.

Liyan Chen, Yael Tauman Kalai, and Zoe Xi. How to avoid debate: Scalable AI safety via doubly-efficient interactive proofs. In *Proceedings of the 43rd International Conference on Machine Learning*, ICML '26, 2026. Full version: arXiv:2607.03561.

Megan Chen, Alessandro Chiesa, and Nicholas Spooner. On succinct non-interactive arguments in relativized worlds. In *Proceedings of the 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT '22, pp. 336–366, 2022.

Megan Chen, Alessandro Chiesa, Tom Gur, Jack O'Connor, and Nicholas Spooner. Proof-carrying data from arithmetized random oracles. In *Proceedings of the 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT '23, pp. 379–404, 2023.

Alessandro Chiesa and Siqi Liu. On the impossibility of probabilistic proofs in relativized worlds. In *Proceedings of the 11th Innovations in Theoretical Computer Science Conference*, ITCS '20, pp. 57:1–57:30, 2020.

Alessandro Chiesa and Eran Tromer. Proof-carrying data and hearsay arguments from signature cards. In *Proceedings of the 1st Symposium on Innovations in Computer Science*, ICS '10, pp. 310–331, 2010.

Alessandro Chiesa, Ziyi Guan, Shahar Samocha, and Eylon Yogev. Security bounds for proof-carrying data from straightline extractors. In *Proceedings of the 22nd Theory of Cryptography Conference*, TCC '24, pp. 464–496, 2024.

Paul Christiano, Buck Shlegeris, and Dario Amodei. Supervising strong learners by amplifying weak experts, 2018. arXiv preprint arXiv:1810.08575.

Thomas Claburn. Chat at your own risk! Data brokers are selling deeply personal bot transcripts. The Register, `https://www.theregister.com/2026/03/03/chatbot_data_harvesting_personal_info/`, March 2026.

Victor Costan and Srinivas Devadas. Intel SGX explained, 2016. IACR Cryptology ePrint Archive, Report 2016/086.

Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, STOC '90, pp. 416–426, 1990.

Olive Franzese, Ali Shahin Shamsabadi, Carter Luck, and Hamed Haddadi. Secure and confidential certificates of online fairness. In *Advances in Neural Information Processing Systems 38*, NeurIPS '25, pp. 40077–40107, 2025.

Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor W. Coley. Sample efficiency matters: A benchmark for practical molecular optimization. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022), Datasets and Benchmarks Track*, 2022.

Sanjam Garg, Aarushi Goel, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, Guru-Vamsi Policharla, and Mingyuan Wang. Experimenting with zero-knowledge proofs of training. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, CCS '23, pp. 1880–1894, 2023. doi: 10.1145/3576915.3623202.

Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989. Preliminary version appeared in STOC '85.

Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: Interactive proofs for muggles. *Journal of the ACM*, 62(4):27:1–27:64, 2015.

Shai Halevi and Silvio Micali. Practical and provably-secure commitment schemes from collision-free hashing. In *Proceedings of the 16th Annual International Cryptology Conference*, CRYPTO '96, pp. 201–215, 1996.

Geoffrey Irving and Simon Marshall. The need to relativise in debate. AI Alignment Forum post, 2025. URL https://www.alignmentforum.org/posts/XycoFucvxAPhcgJQa/the-need-to-relativise-in-debate-1.

Geoffrey Irving, Paul Christiano, and Dario Amodei. AI safety via debate, 2018. arXiv preprint arXiv:1805.00899.

Yuval Ishai and Mor Weiss. Probabilistically checkable proofs of proximity with zero-knowledge. In *Proceedings of the 11th Theory of Cryptography Conference*, TCC '14, pp. 121–145, 2014.

Madura K. P. Jayatunga, Margaret Ayers, Lotte Bruens, Dhruv Jayanth, and Christoph Meier. How successful are AI-discovered drugs in clinical trials? A first analysis and emerging lessons. *Drug Discovery Today*, 29(6):104009, 2024. doi: 10.1016/j.drudis.2024.104009.

Jonathan Katz. Universally composable multi-party computation using tamper-proof hardware. In *Proceedings of the 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT '07, pp. 115–128, 2007. doi: 10.1007/978-3-540-72540-4_7.

Joe Kilian. A note on efficient zero-knowledge proofs and arguments. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, STOC '92, pp. 723–732, 1992.

Jan Hendrik Kirchner, Yining Chen, Harri Edwards, Jan Leike, Nat McAleese, and Yuri Burda. Prover–verifier games improve legibility of LLM outputs, 2024. arXiv preprint arXiv:2407.13692.

Ravie Lakshmanan. Claude extension flaw enabled zero-click XSS prompt injection via any website. The Hacker News, https://thehackernews.com/2026/03/claude-extension-flaw-enabled-zero.html, March 2026.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: A research direction, 2018. arXiv preprint arXiv:1811.07871.

Tianyi Liu, Xiang Xie, and Yupeng Zhang. zkCNN: Zero knowledge proofs for convolutional neural network predictions and accuracy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, pp. 2968–2985, 2021. doi: 10.1145/3460120.3485379.

Mary Maller, Nicolas Mohnblatt, and Arantxa Zapico. IVC in the open-and-sign random oracle model. In *Proceedings of the 31st International Conference on the Theory and Application of Cryptology and Information Security*, ASIACRYPT '25, pp. 333–365, 2025.

Iain Martin and Emily Baker-White. Elon Musk's xAI published hundreds of thousands of Grok chatbot conversations. Forbes, https://www.forbes.com/sites/iainmartin/2025/08/20/elon-musks-xai-published-hundreds-of-thousands-of-grok-chatbot-conversations/, August 2025.

Frank McKeen, Ilya Alexandrovich, Alex Berenzon, Carlos V. Rozas, Hisham Shafi, Vedvyas Shanbhogue, and Uday R. Savagaonkar. Innovative instructions and software model for isolated execution. In *Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy*, HASP '13, 2013. doi: 10.1145/2487726.2488368.

Fan Mo, Hamed Haddadi, Kleomenis Katevas, Eduard Marin, Diego Perino, and Nicolas Kourtellis. PPFL: Privacy-preserving federated learning with trusted execution environments. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '21, pp. 94–108, 2021. doi: 10.1145/3458864.3466628.

Michael Moor, Oishi Banerjee, Zahra Shakeri Hossein Abad, Harlan M. Krumholz, Jure Leskovec, Eric J. Topol, and Pranav Rajpurkar. Foundation models for generalist medical artificial intelligence. *Nature*, 616(7956):259–265, 2023. doi: 10.1038/s41586-023-05881-4.

Gal Nagli. Wiz research uncovers exposed DeepSeek database leaking sensitive information, including chat history. Wiz Research, `https://www.wiz.io/blog/wiz-research-uncovers-exposed-deepseek-database-leak`, January 2025.

Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, STOC '89, pp. 33–43, 1989.

Shakked Noy and Whitney Zhang. Experimental evidence on the productivity effects of generative artificial intelligence. *Science*, 381(6654):187–192, 2023. doi: 10.1126/science.adh2586.

Martina Padovan, Bianca Cosci, Armando Petillo, Gianluca Nerli, Francesco Porciatti, Sergio Scarinci, Francesco Carlucci, Letizia Dell'Amico, Niccolò Meliani, Gabriele Necciari, Vincenzo Carmelo Lucisano, Riccardo Marino, Rudy Foddis, and Alessandro Palla. ChatGPT in occupational medicine: A comparative study with human experts. *Bioengineering*, 11(1):57, 2024. doi: 10.3390/bioengineering11010057.

Rafael Pass, Elaine Shi, and Florian Tramèr. Formal abstractions for attested execution secure processors. In *Proceedings of the 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT '17, pp. 260–289, 2017. doi: 10.1007/978-3-319-56620-7_10.

Christian Priebe, Kapil Vaswani, and Manuel Costa. EnclaveDB: A secure database using SGX. In *Proceedings of the 39th IEEE Symposium on Security and Privacy*, S&P '18, pp. 264–278, 2018. doi: 10.1109/SP.2018.00025.

Itay Ravia. Breaking down EchoLeak, the first zero-click AI vulnerability enabling data exfiltration from Microsoft 365 Copilot. `https://www.catonetworks.com/blog/breaking-down-echoleak/`, May 2025.

John Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, STOC '90, pp. 387–394, 1990.

Antony Rowstron. Agentic witnessing: Pragmatic and scalable TEE-enabled privacy-preserving auditing, 2026. arXiv:2604.24203.

Will Shanklin. OpenAI is removing ChatGPT conversations from Google. Engadget, `https://www.engadget.com/ai/openai-is-removing-chatgpt-conversations-from-google-194735704.html`, August 2025.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Abubakr Babiker, Nathanael Schärli, Aakanksha Chowdhery, Philip Mansfield, Dina Demner-Fushman, Blaise Agüera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023. doi: 10.1038/s41586-023-06291-2.

Jonathan M. Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M. Donghia, Craig R. MacNair, Shawn French, Lindsey A. Carfrae, Zohar Bloom-Ackermann, Victoria M. Tran, Anush Chiappino-Pepe, Ahmed H. Badran, Ian W. Andrews, Emma J. Chory, George M. Church, Eric D. Brown, Tommi S. Jaakkola, Regina Barzilay, and James J. Collins. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702, 2020. doi: 10.1016/j.cell.2020.01.021.

Haochen Sun, Jason Li, and Hongyang Zhang. zkLLM: Zero knowledge proofs for large language models. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, CCS '24, pp. 4405–4419, 2024. doi: 10.1145/3658644.3670334.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature Medicine*, 29(8): 1930–1940, 2023. doi: 10.1038/s41591-023-02448-8.

Florian Tramèr and Dan Boneh. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. In *Proceedings of the 7th International Conference on Learning Representations*, ICLR '19, 2019.

Paul Valiant. Incrementally verifiable computation or proofs of knowledge imply time/space efficiency. In *Proceedings of the 5th Theory of Cryptography Conference*, TCC '08, pp. 1–18, 2008.

Francesca Vella, Paola Senia, Veronica Filetti, Ermanno Vitale, Giuseppe Sferrazzo, Lorenzo Privitera, Venerando Rapisarda, Giuliano Indelicato, Nektaria Zagorianakou, Stefania Durante, and Lucia Rapisarda. Application of large language models as decision support tools in occupational health and safety management: a cohort study of industrial workers. *Frontiers in Public Health*, 14: 1815316, 2026. doi: 10.3389/fpubh.2026.1815316.

Suppakit Waiwitlikhit, Ion Stoica, Yi Sun, Tatsunori Hashimoto, and Daniel Kang. Trustless audits without revealing data or models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML '24*, pp. 49808–49821, 2024.

Felix Wong, Erica J. Zheng, Jacqueline A. Valeri, Nina M. Donghia, Melis N. Anahtar, Satotaka Omori, Alicia Li, Andres Cubillos-Ruiz, Aarti Krishnan, Wengong Jin, Abigail L. Manson, Jens Friedrichs, Ralf Helbig, Behnoush Hajian, Dawid K. Fiejtek, Florence F. Wagner, Holly H. Soutter, Ashlee M. Earl, Jonathan M. Stokes, Lars D. Renner, and James J. Collins. Discovery of a structural class of antibiotics with explainable deep learning. *Nature*, 626(7997):177–185, 2024. doi: 10.1038/s41586-023-06887-8.

Zuojun Xu, Feng Ren, Ping Wang, Jie Cao, Chunting Tan, Dedong Ma, Li Zhao, Jinghong Dai, Yipeng Ding, Haohui Fang, Huiping Li, Hong Liu, Fengming Luo, Ying Meng, Pinhua Pan, Pingchao Xiang, Zuke Xiao, Sujata Rao, Carol Satler, Sang Liu, Yuan Lv, Heng Zhao, Shan Chen, Hui Cui, Mikhail Korzinkin, David Gennert, and Alex Zhavoronkov. A generative AI-discovered TNIK inhibitor for idiopathic pulmonary fibrosis: a randomized phase 2a trial. *Nature Medicine*, 31(8):2602–2610, 2025. doi: 10.1038/s41591-025-03743-2.

Alex Zhavoronkov, Yan A. Ivanenkov, Alex Aliper, Mark S. Veselov, Vladimir A. Aladinskiy, Anastasiya V. Aladinskaya, Victor A. Terentiev, Daniil A. Polykovskiy, Maksim D. Kuznetsov, Arip Asadulaev, Yury Volkov, Artem Zholus, Rim R. Shayakhmetov, Alexander Zhebrak, Lidiya I. Minaeva, Bogdan A. Zagribelnyy, Lennart H. Lee, Richard Soll, David Madge, Li Xing, Tao Guo, and Alán Aspuru-Guzik. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nature Biotechnology*, 37(9):1038–1040, 2019. doi: 10.1038/s41587-019-0224-x.

A PRELIMINARIES

Some of our definitions in this section are taken verbatim from Barbara et al. (2025).

For a set S, we denote by 2^S the set of all subsets of S. All sets considered in this paper are finite. For every finite set, we can define a bijective binary encoding. Therefore, without loss of generality, we assume every set considered in this paper is a subset of $\{0,1\}^*$.

We write $x \leftarrow \mathcal{D}$ to denote that x is a sample of the distribution $\mathcal{D}$. We write $x \leftarrow \mathcal{X}$ to denote that x is sampled uniformly at random from the non-empty set $\mathcal{X}$. We write $x \in \mathcal{D}$ to denote that x is in the support of $\mathcal{D}$, i.e. the value x is sampled with non-zero probability from $\mathcal{D}$.

A.1 ORACLES, LANGUAGES AND RELATIONS

Definition A.1. *For all finite sets X and Y, the set $\mathsf{Func}(X,Y)$ is the set of all functions $f\colon X \to Y$.*

Definition A.2 (Oracle)**.** *An* **oracle** *is a collection $\mathcal{O} = \{\mathcal{O}_{\lambda,n}\}_{\lambda\in\mathbb{N},n\in\mathbb{N}}$ where each $\mathcal{O}_{\lambda,n}$ is a distribution over functions $\mathsf{Func}(X_{\lambda,n},Y_{\lambda,n})$ for some finite sets $X_{\lambda,n},Y_{\lambda,n}$ such that the empty string $\perp \notin X_{\lambda,n}$.*

We write $f \in \mathcal{O}$ to denote that f is in the support of $\mathcal{O}_{\lambda,n}$ for some $\lambda \in \mathbb{N}, n \in \mathbb{N}$. An oracle $\mathcal{O}$ has **input size bound** ℓ_{in} *if for every $\lambda, n \in \mathbb{N}$, $X_{\lambda,n} \subseteq \{0,1\}^{\leq \ell_{\mathsf{in}}(\lambda,n)}$.*

The empty oracle $O_\perp$ is the oracle that the only function $f \in O_\perp$ is the function that the domain is the empty set.

Definition A.3 (Relativized language)**.** *Let $\mathcal{O}$ be an oracle. A relativized language $L_{\mathcal{O}}$ is a collection of languages indexed by functions $f \in \mathcal{O}_{\lambda,n}$ for $\lambda, n \in \mathbb{N}$, namely,*

$$L_{\mathcal{O}} \coloneqq \{L_f\}_{\lambda,n\in\mathbb{N},f\in\mathcal{O}_{\lambda,n}} \ .$$

Definition A.4 (Relativized relation)**.** *Let $\mathcal{O}$ be an oracle. A relativized relation $R_{\mathcal{O}}$ is a collection of relations indexed by a security bound $\lambda \in \mathbb{N}$ and instance size $n \in \mathbb{N}$ and functions $f \in \mathcal{O}_{\lambda,n}$, namely,*

$$R_{\mathcal{O}} \coloneqq \{R_f\}_{\lambda,n\in\mathbb{N},f\in\mathcal{O}_{\lambda,n}} \ .$$

The relativized language corresponding to the relativized relation $R_{\mathcal{O}}$, denoted $L(R_{\mathcal{O}})$, is defined as follows:

$$L(R_{\mathcal{O}}) \coloneqq \{L(R_f)\}_{\lambda,n\in\mathbb{N},f\in\mathcal{O}_{\lambda,n}} \ ,$$

where each $L(R_f) \coloneqq \{\mathbb{x} : \exists\, \mathbb{w} \text{ s.t. } (\mathbb{x},\mathbb{w}) \in R_f\}$.

Remark A.5. Let R be a relation and $\mathcal{O}$ be an oracle. We can view R as a relativized relation $\{R_f\}_{f\in\mathcal{O}}$ where $R_f \coloneqq R$ for every $f \in \mathcal{O}$.

Definition A.6. *Let $R_{\mathcal{O}}$ be a relativized relation. For $f \in \mathcal{O}$ and $\mathbb{x} \in \{0,1\}^*$, the* **set of witnesses for $\mathbb{x}$ in** R_f *is* $\mathsf{Witness}^{R_{\mathcal{O}}}(f,\mathbb{x}) \coloneqq \{\mathbb{w} \in \{0,1\}^* : (\mathbb{x},\mathbb{w}) \in R_f\}$.

Definition A.7. *For every family of pairs of sets $((X_{\lambda,n},Y_{\lambda,n}))_{\lambda,n\in\mathbb{N}}$, the* **random oracle over** $((X_{\lambda,n},Y_{\lambda,n}))_{\lambda,n\in\mathbb{N}}$ *is the collection $\{\mathcal{U}_{\lambda,n}\}_{\lambda,n\in\mathbb{N}}$, where each $\mathcal{U}_{\lambda,n}$ is the uniform distribution over $\mathsf{Func}(X_{\lambda,n},Y_{\lambda,n})$.*

Definition A.8. *An oracle $\mathcal{O}$ is a* **uniform oracle over** $((X_{\lambda,n},Y_{\lambda,n}))_{\lambda,n\in\mathbb{N}}$ *if for every $\lambda, n \in \mathbb{N}$, $\mathcal{O}_{\lambda,n}$ is the uniform distribution over a subset of $\mathsf{Func}(X_{\lambda,n},Y_{\lambda,n})$ for some finite sets $X_{\lambda,n},Y_{\lambda,n}$.*

Remark A.9. The random oracle is a uniform oracle.

A.2 TURING MACHINES AND COMPLEXITY CLASSES WITH ORACLES

To define the complexity class $\mathsf{NTIME}^{\mathcal{O}}[t]$, we consider Turing machines that can query oracles. An oracle machine is a machine that has *black-box* access to a function $f : X \to Y$ that the machine can query. Each query costs the machine a single computational step.

Definition A.10. *Let $\mathcal{O}$ be an oracle. An* **oracle Turing machine** M *is a Turing machine that has two additional special tapes, the* oracle query tape *and the* oracle answer tape*, and two additional special states, the* QUERY *state and the* ANSWER *state. Given input $\mathbb{x} \in \{0,1\}^*$ and oracle access to a function $f \in \mathcal{O}_{\lambda,n}$ for some λ, n, $M^f(\mathbb{x})$ works as follows:*

- *The input $\mathbb{x}$ is written in the input tape.*
- *The execution of M proceeds as a normal Turing machine computation except that, when M intends to make a query to the oracle, then it enters the* QUERY *state.*
- *In the* QUERY *state, M writes its query $q \in \{0,1\}^*$ on the query tape.*
- *In the following step, M enters the* ANSWER *state and the content in the answer tape becomes $f(q)$ if q is in the domain of f, becomes (n, λ) if $q = \perp$, and empty otherwise.*

We call the function that maps the tuple of the input x and the oracle answers to the first $i-1$ oracle queries to M's i-th query the i-th query function of M.

We extend the definition of oracle Turing machines to deterministic oracle Turing machines, nondeterministic oracle Turing machines, and probabilistic oracle Turing machines with the standard definitions of deterministic Turing machines, nondeterministic Turing machines, and probabilistic Turing machines in prior literature. Throughout this paper, we focus on oracle algorithms, which are implemented as oracle Turing machines.

Definition A.11. *The* **view** *of a probabilistic interactive oracle algorithm A is a function View_A that maps the oracle f, the input z, the messages $(a_i)_{i\in[k]}$ that the algorithm receives, and the randomness r to the tuple $(z, r, (a_i)_{i\in[k]}, \mu := ((q_i, f(q_i)))_{i\in[\mathsf{q}]})$, where $(q_i)_{i\in[\mathsf{q}]}$ is the list of all oracle queries of $A^f(z), A^f(z, a_1), \ldots,$ and $A^f(z, a_1, \ldots, a_k)$.*

For every oracle f, input z, messages $(a_i)_{i\in[k]}$ and randomness r, the output of $A^f(z, (a_i)_{i\in[k]}; r)$ is a function of $\mathsf{View}_A(f, z, (a_i)_{i\in[k]}, r)$.

Definition A.12. *Let A_0 and A_1 be interactive oracle algorithms, interacting for k rounds, with respective input domains Z_0 and Z_1 and randomness complexities ar_0 and ar_1. For $f_0, f_1 \in \mathcal{O}$, $z_0 \in Z_0$, $z_1 \in Z_1$, we denote the random experiment*

$$\left\{ \begin{array}{l} r_0 \leftarrow \{0,1\}^{\mathsf{ar}_0} \\ r_1 \leftarrow \{0,1\}^{\mathsf{ar}_1} \\ \text{For } i \in [\mathsf{k}]: \\ \quad b_i := A_0^{f_0}(z_0, (a_j)_{j<i}; r_0) \\ \quad a_i := A_1^{f_1}(z_1, (b_j)_{j\le i}; r_1) \\ \mathsf{d} := a_{\mathsf{k}} \\ \mathsf{tr} := (b_1, a_1, \ldots, b_{\mathsf{k}}, a_{\mathsf{k}}) \end{array} \right\}$$

by $\mathsf{d} \xleftarrow{r_0, r_1, \mathsf{tr}} \langle A_0^{f_0}(z_0), A_1^{f_1}(z_1)\rangle$. When r_0, r_1, tr are irrelevant, we omit them from the notation.

If $\mathsf{ar}_0 = \mathsf{ar}_1 = 0$, we write $\mathsf{d} \xleftarrow{\mathsf{tr}} \langle A_0^{f_0}(z_0), A_1^{f_1}(z_1)\rangle$.

Nondeterministic time. A nondeterministic machine M is an ***ND-decider*** for a relation R if for every $(\mathbb{x}, \mathbb{w})$ it holds that $M(\mathbb{x}, \mathbb{w}) = 1$ if and only if $(\mathbb{x}, \mathbb{w}) \in R$, where $M(\mathbb{x}, \mathbb{w})$ denotes the decision of M on the input $\mathbb{x}$ on the branch $\mathbb{w}$.

Definition A.13 (Relativized NTIME)**.** *Let $\mathcal{O}$ be an oracle. Let t be a time-constructible function. The complexity class $\mathsf{NTIME}^{\mathcal{O}}[t]$ is the set of relativized relations $R_{\mathcal{O}} = \{R_f\}_{\lambda,n\in\mathbb{N}, f\in\mathcal{O}_{\lambda,n}}$ such that there exists an oracle nondeterministic machine M that runs in time $O(t(n))$ and, for every $\lambda, n \in \mathbb{N}$,*

$$\Pr\left[\, M^f \text{ is an ND-decider for } R_f \;\middle|\; f \leftarrow \mathcal{O}_{\lambda,n} \,\right] = 1 \ .$$

We call M an **ND-decider** *for $R_{\mathcal{O}}$.*

A.3 Relativized interactive arguments in an oracle model

Let $\mathcal{O}$ be an oracle. A **relativized interactive argument** in the oracle model $\mathcal{O}$ for a relativized relation $R_{\mathcal{O}}$ is a tuple of interactive oracle algorithms $\mathsf{ARG} = (\mathcal{P}, \mathcal{V})$ that works as follows: Given security parameter $\lambda \in \mathbb{N}$, instance size $n \in \mathbb{N}$, oracle instance $f \in \mathcal{O}_{\lambda,n}$, instance $\mathbb{x} \in \{0,1\}^n$ and witness $\mathbb{w} \in \{0,1\}^*$,

1. Sample the verifier randomness $\rho \leftarrow \{0,1\}^{\mathsf{vr}}$.
2. Sample the prover randomness $\zeta \leftarrow \{0,1\}^{\mathsf{pr}}$.
3. For $i \in [\mathsf{k}]$:

(a) The prover $\mathcal{P}^f(1^\lambda, (\mathbb{x}, \mathbb{w}), (a_j)_{j<i}; \zeta)$ computes the i-th message b_i.
(b) The verifier $\mathcal{V}^f(1^\lambda, \mathbb{x}, (b_j)_{j\le i}; \rho)$ computes the i-th message a_i.

4. The verifier $\mathcal{V}^f(1^\lambda, \mathbb{x}, (b_j)_{j\in[\mathsf{k}]}; \rho)$ outputs the decision $\mathsf{d} := a_{\mathsf{k}} \in \{0,1\}$.

Definition A.14 (Completeness)**.** *For every security parameter $\lambda \in \mathbb{N}$, instance size $n \in \mathbb{N}$, instance $\mathbb{x} \in \{0,1\}^n$, and adversary $\mathcal{A}$,*

$$\Pr\left[\begin{array}{l|l} \mathsf{d} = 0 \\ \wedge\, (\mathbb{x}, \mathbb{w}) \in R_f & \begin{array}{l} f \leftarrow \mathcal{O}_{\lambda,n} \\ \mathbb{w} \leftarrow \mathcal{A}^f(\mathbb{x}) \\ \mathsf{d} \leftarrow \langle \mathcal{P}^f(1^\lambda, (\mathbb{x}, \mathbb{w})), \mathcal{V}^f(1^\lambda, \mathbb{x})\rangle \end{array} \end{array}\right] \le \alpha(\lambda, n) \ .$$

Definition A.15 (Soundness)**.** *For every security parameter $\lambda \in \mathbb{N}$, instance size bound $n \in \mathbb{N}$, instance $\mathbb{x} \in \{0,1\}^n$, adversary query bound $\mathsf{q}_{\widetilde{\mathcal{P}}} \in \mathbb{N}$, adversary time bound $\mathsf{t}_{\widetilde{\mathcal{P}}} \in \mathbb{N}$, and $\mathsf{q}_{\widetilde{\mathcal{P}}}$-query $\mathsf{t}_{\widetilde{\mathcal{P}}}$-time adversary $\widetilde{\mathcal{P}}$,*

$$\Pr\left[\begin{array}{l|l} \mathsf{d} = 1 \\ \wedge\, \mathbb{x} \notin L(R_f) & \begin{array}{l} f \leftarrow \mathcal{O}_{\lambda,n} \\ \mathsf{d} \leftarrow \langle \widetilde{\mathcal{P}}^f(\mathbb{x}), \mathcal{V}^f(1^\lambda, \mathbb{x})\rangle \end{array} \end{array}\right] \le \beta(\lambda, n, \mathsf{q}_{\widetilde{\mathcal{P}}}, \mathsf{t}_{\widetilde{\mathcal{P}}}) \ .$$

Efficiency measures. We consider several efficiency measures of ARG: for every security parameter $\lambda \in \mathbb{N}$, instance size bound $n \in \mathbb{N}$, and oracle function $f \in \mathcal{O}_{\lambda,n}$,

- the *round complexity* k is the maximum number of rounds the prover $\mathcal{P}$ and the verifier $\mathcal{V}$ are allowed to interact;
- the *prover communication* $\mathsf{pc}(\lambda, n)$ is $\sum_{i\in[\mathsf{k}]} \mathsf{pc}_i(\lambda, n)$, where $\mathsf{pc}_i(\lambda, n)$ is the number of bits the prover $\mathcal{P}$ is allowed to send to the verifier $\mathcal{V}$ in round i for $i \in [\mathsf{k}]$;
- the *verifier communication* $\mathsf{vc}(\lambda, n)$ is the maximum number of bits the verifier $\mathcal{V}$ is allowed to send to the prover $\mathcal{P}$;
- the *verifier query complexity* $\mathsf{vq}(\lambda, n)$ is the maximum number of queries to the oracle by the verifier $\mathcal{V}$;
- the *verifier running time* $\mathsf{vt}(\lambda, n)$ is the maximum number of operations performed by the verifier $\mathcal{V}$;
- the *honest prover query complexity* $\mathsf{pq}(\lambda, n)$ is the maximum number of queries to the oracle by the prover $\mathcal{P}$;
- the *honest prover time* $\mathsf{pt}(\lambda, n)$ is the maximum number of operations performed by the prover $\mathcal{P}$;
- the *verifier randomness complexity* $\mathsf{vr}(\lambda, n)$ is the number of bits of randomness used by the argument verifier $\mathcal{V}$;
- the *honest prover randomness complexity* $\mathsf{pr}(\lambda, n)$ is the number of bits of randomness used by the argument prover $\mathcal{P}$.

Definition A.16 (Relativized ARG in an oracle model)**.** *Let $\mathcal{O}$ be an oracle. The complexity class*

$$\mathsf{ARG}^{\mathcal{O}}\left[\begin{array}{lcl} \text{completeness error} & \alpha & = \alpha(\lambda, n) \\ \text{soundness error} & \beta & = \beta(\lambda, n, \mathsf{q}_{\widetilde{\mathcal{P}}}, \mathsf{t}_{\widetilde{\mathcal{P}}}) \\ \text{verifier query bound} & \mathsf{vq} & = \mathsf{vq}(\lambda, n) \\ \text{verifier running time} & \mathsf{vt} & = \mathsf{vt}(\lambda, n) \\ \text{prover communication} & \mathsf{pc} & = \mathsf{pc}(\lambda, n) \\ \text{honest prover query bound} & \mathsf{pq} & = \mathsf{pq}(\lambda, n) \\ \text{honest prover time bound} & \mathsf{pt} & = \mathsf{pt}(\lambda, n) \end{array}\right]$$

is the set of relativized relations $R_{\mathcal{O}} = \{R_f\}_{\lambda,n\in\mathbb{N},f\in\mathcal{O}_{\lambda,n}}$ that admit a relativized interactive argument $(\mathcal{P}, \mathcal{V})$ such that the following holds:

- *$(\mathcal{P}, \mathcal{V})$ has completeness error α and soundness error β;*
- *the verifier query complexity is vq;*
- *the verifier running time is vt;*
- *the prover communication is pc;*
- *the honest prover query complexity is pq;*
- *the honest prover running time is pt.*

Definition A.17. *Let $\mathsf{ARG} = (\mathcal{P}, \mathcal{V})$ be a relativized interactive argument for a relativized relation $R_{\mathcal{O}}$. ARG has* **witness-hiding error** $\varepsilon_{\mathsf{WH}}$ *if for every security parameter $\lambda \in \mathbb{N}$, instance size bound*

$n \in \mathbb{N}$, *instance* $\mathbb{x} \in \{0,1\}^n$, *adversary* $\mathcal{A}_{\mathsf{WH},1}$, *and* $\mathsf{aq}(\lambda, n)$*-query* $\mathsf{at}(\lambda, n)$*-time adversary* $\mathcal{A}_{\mathsf{WH},2}$, *the following is at most* $\varepsilon_{\mathsf{WH}}(\lambda, n, \mathsf{aq}, \mathsf{at})$.

$$\Pr\left[(\mathbb{x}, \mathbb{w}') \in R_f \ \wedge\ (\mathbb{x}, \mathbb{w}) \in R_f \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_{\lambda,n} \\ \mathbb{w} \leftarrow \mathcal{A}^f_{\mathsf{WH},1}(\mathbb{x}) \\ \mathsf{d} \xleftarrow{\zeta,\rho,\mathsf{tr}} \langle \mathcal{P}^f(1^\lambda, (\mathbb{x}, \mathbb{w})), \mathcal{V}^f(1^\lambda, \mathbb{x}) \rangle \\ \mathbb{w}' \leftarrow \mathcal{A}^f_{\mathsf{WH},2}(\mathbb{x}, \rho, \mathsf{tr}) \end{array} \right] .$$

Definition A.18 (Black-box rewinding access)**.** *Let* $\mathcal{A}$ *be a probabilistic interactive oracle algorithm and* f *a fixed oracle. An algorithm* $\mathcal{B}$ *has* **black-box rewinding access** *to* $\mathcal{A}$ *relative to* f *if it may access executions of* $\mathcal{A}$ *only as follows.*

1. **Fresh execution.** *$\mathcal{B}$ may start a fresh execution of $\mathcal{A}$ on a fixed input and an auxiliary input. Its random tape is sampled independently according to $\mathcal{A}$'s randomness distribution and is hidden from $\mathcal{B}$.*
2. **Interaction.** *$\mathcal{B}$ may exchange messages with an active execution. Every oracle query of $\mathcal{A}$ is answered by the same fixed oracle f.*
3. **Rewinding.** *$\mathcal{B}$ may checkpoint and later restore an active execution. A checkpoint is opaque, and restoration recovers the exact saved configuration, including the random tape and its position. Thus every continuation from a checkpoint uses the same random tape, including the same unread coins.*
4. **View.** *After an execution terminates, $\mathcal{B}$ may obtain its complete view as defined in Definition A.11. Once this view is obtained, that execution and its checkpoints may no longer be restored or continued.*

All executions, including rewound or discarded continuations, and all oracle queries made in them count toward the resource bounds of $\mathcal{B}$. If $\mathcal{A}$ has no oracle access, the oracle calls are omitted.

Definition A.19. *A relativized argument $(\mathcal{P}, \mathcal{V})$ for a relativized relation $R_{\mathcal{O}}$ has* **malicious verifier zero-knowledge error** *z* **with simulator time** st **and query complexity** sq *if there exists a* st*-time* sq*-query simulator $\mathcal{S}$ with black-box rewinding access to the malicious verifier as in Definition A.18 such that for every $\lambda, n \in \mathbb{N}$, for every $f \in \mathcal{O}_{\lambda,n}$, for every $(\mathbb{x}, \mathbb{w}) \in R_f$, and* at*-time* aq*-query malicious verifier $\widetilde{\mathcal{V}}$, the following distributions are $z(\lambda, n, \mathsf{aq}, \mathsf{at})$-close in statistical distance:*

$$\{\mathcal{S}^{f,\widetilde{\mathcal{V}}}(1^\lambda, \mathbb{x})\}$$

$$\textit{and} \quad \left\{ \mathsf{View}_{\widetilde{\mathcal{V}}}(f, \mathbb{x}, (b_i)_{i\in[k]}, r) \;\middle|\; \mathsf{d} \xleftarrow{\zeta,r,(b_1,a_1,\ldots,b_k,a_k)} \langle \mathcal{P}^f(1^\lambda, (\mathbb{x}, \mathbb{w})), \widetilde{\mathcal{V}}^f(\mathbb{x}) \rangle \right\} ,$$

where $\mathsf{View}_{\widetilde{\mathcal{V}}}(f, \mathbb{x}, (b_i)_{i\in[k]}, r)$ *is $\widetilde{\mathcal{V}}$'s view as in Definition A.11.*

Definition A.20. *Let* $\mathsf{ARG} = (\mathcal{P}, \mathcal{V})$ *be a relativized interactive argument for a relativized relation $R_{\mathcal{O}}$.* ARG *has* **knowledge soundness error** $\kappa = \{\kappa_f\}_{f\in\mathcal{O}}$ **with extractor time** et *if there exists a probabilistic extractor $\mathcal{E}$ such that the following holds. For every security parameter $\lambda \in \mathbb{N}$, instance size bound $n \in \mathbb{N}$, $f \in \mathcal{O}_{\lambda,n}$, instance $\mathbb{x} \in \{0,1\}^n$, and* aq*-query* at*-time malicious prover $\widetilde{\mathcal{P}}$, if*

$$\delta := \Pr\left[\mathsf{d} = 1 \;\middle|\; \mathsf{d} \leftarrow \langle \widetilde{\mathcal{P}}^f(1^\lambda, \mathbb{x}), \mathcal{V}^f(1^\lambda, \mathbb{x}) \rangle \right] - \kappa_f(\lambda, |\mathbb{x}|, \mathsf{aq}, \mathsf{at}) > 0 ,$$

then for every $0 < \delta' \le \delta$, the extractor $\mathcal{E}^{f,\widetilde{\mathcal{P}}}(1^\lambda, \mathbb{x}, \delta')$, with black-box rewinding access to the malicious prover $\widetilde{\mathcal{P}}$, as in Definition A.18, outputs a witness $\mathbb{w}$ satisfying $(\mathbb{x}, \mathbb{w}) \in R_f$ with probability at least $2/3$, and runs in worst-case time $\mathsf{et}(\lambda, n, \mathsf{at}, \delta')$.

A.4 Digital signatures

A digital signature scheme is defined as follows.

Definition A.21. *A* **digital signature scheme** *for messages in $\{0,1\}^*$ is a tuple of polynomial-time algorithms* $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *with the following syntax.*

- $\mathsf{Gen}(1^\lambda)$ *outputs a signing key* sk *and a verification key* vk.
- $\mathsf{Sign}(\mathsf{sk}, m)$ *outputs a signature σ on a message m.*
- $\mathsf{Verify}(\mathsf{vk}, m, \sigma)$ *deterministically outputs a bit indicating whether σ is a valid signature on m under* vk.

The scheme SIG *is a digital signature scheme with unforgeability error* $\varepsilon_{\mathsf{sig}}$ *if the following hold.*

- **Perfect correctness.** *For every* $\lambda \in \mathbb{N}$*, every* $(\mathsf{sk}, \mathsf{vk}) \in \mathsf{Supp}(\mathsf{Gen}(1^\lambda))$*, and every message* m*,*
$$\Pr\left[\, \mathsf{Verify}(\mathsf{vk}, m, \sigma) = 1 \mid \sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m) \,\right] = 1 \ .$$
- **Existential unforgeability under adaptive chosen-message attacks (EUF-CMA) error** $\varepsilon_{\mathsf{sig}}$ **relative to** $\mathcal{O}$**.** *For every* $f \in \mathcal{O}$*,* t*-time adversary* $\mathcal{A}^f$ *making at most* q *queries in total to* f *and to the adaptive signing oracle* $\mathsf{Sign}(\mathsf{sk}, \cdot)$*,*
$$\Pr\left[\mathsf{Verify}(\mathsf{vk}, m^\star, \sigma^\star) = 1 \ \wedge \ m^\star \notin Q \;\middle|\; \begin{array}{l} (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^\lambda) \\ (m^\star, \sigma^\star) \leftarrow \mathcal{A}^{f, \mathsf{Sign}(\mathsf{sk}, \cdot)}(1^\lambda, \mathsf{vk}) \end{array} \right] \leq \varepsilon^f_{\mathsf{sig}}(\lambda, \mathsf{q}, \mathsf{t}) \ ,$$
where Q *is the set of messages submitted to the signing oracle* $\mathsf{Sign}(\mathsf{sk}, \cdot)$*. The scheme is* **EUF-CMA secure relative to** $\mathcal{O}$ *if for every* $f \in \mathcal{O}_{\lambda,n}$*,* $\varepsilon^f_{\mathsf{sig}}(\lambda, \mathsf{q}, \mathsf{t})$ *is negligible whenever* $n, \mathsf{q}, \mathsf{t}$ *are polynomially bounded in* λ*.*

Theorem A.22 ((Naor & Yung, 1989; Rompel, 1990))**.** *Assuming the existence of one-way functions (OWF), there exists an EUF-CMA secure digital signature scheme (relative to the empty oracle).*

Since OWFs can be constructed from collision resistant hash functions, the above theorem extends as follows.

Corollary A.23. *Let* $\mathcal{O}$ *be an oracle. If there exists a hash function family that is collision resistant both in the plain model and relative to* f *for all* $f \in \mathcal{O}$*, then there exists a digital signature scheme that is EUF-CMA secure relative to* $\mathcal{O}$*.*

A.5 Zero-knowledge PCPs of knowledge

The computation bound $T(n)$ is supplied in binary or is computable in time $\mathsf{poly}(n, \log T(n))$.

Definition A.24 (PCP of knowledge)**.** *A binary non-adaptive* **PCP of knowledge** *for an ordinary relation* R *is a triple* $(\mathcal{P}_{\mathsf{pcp}}, \mathcal{V}_{\mathsf{pcp}}, \mathcal{D}_{\mathsf{pcp}})$*. On input* $(\mathbb{x}, \mathbb{w}) \in R$*, the randomized prover outputs* $\pi_{\mathsf{pcp}} \in \{0,1\}^N$*. On input* $\mathbb{x}$*, the verifier samples* $r \leftarrow \{0,1\}^a$*, computes indices* $I_{\mathbb{x},r} \in [N]^k$ *and a predicate* $D_{\mathbb{x},r} \colon \{0,1\}^k \to \{0,1\}$*, and accepts precisely when* $D_{\mathbb{x},r}(\pi_{\mathsf{pcp}}|_{I_{\mathbb{x},r}}) = 1$*. Here restriction to an index tuple retains its order and repetitions. The randomized decoder receives the entire proof string and outputs a candidate witness or* $\perp$*. The parameters may depend on* $n = |\mathbb{x}|$*.*

The PCP has **perfect completeness** *if the verifier accepts every honestly generated proof of a valid instance with probability one. It has* **knowledge threshold** $\kappa_{\mathsf{dec}} \in [0,1)$ *if, for every* $\mathbb{x}$ *and every string* $\pi \in \{0,1\}^N$*,*
$$\mathsf{acc}_{\mathsf{pcp}}(\mathbb{x}, \pi) > \kappa_{\mathsf{dec}} \quad \Longrightarrow \quad \Pr\left[\, (\mathbb{x}, \mathcal{D}_{\mathsf{pcp}}(\mathbb{x}, \pi)) \in R \,\right] \geq \frac{2}{3},$$
where
$$\mathsf{acc}_{\mathsf{pcp}}(\mathbb{x}, \pi) \coloneqq \Pr\left[\, \mathcal{V}^{\pi}_{\mathsf{pcp}}(\mathbb{x}) = 1 \,\right].$$
The probability on the left is over verifier coins; the extraction probability is over independent decoder coins. In particular, false instances have acceptance probability at most κ_{dec} *for every proof string.*

Definition A.25 (Perfect honest-verifier zero knowledge)**.** *A PCP as above is* **perfect honest-verifier zero knowledge** *(HVZK) if a randomized algorithm* $\mathcal{S}_{\mathsf{pcp}}$ *satisfies, for every* $(\mathbb{x}, \mathbb{w}) \in R$*,*
$$\mathcal{S}_{\mathsf{pcp}}(\mathbb{x}) \ \equiv \ \left\{ \left(r, \pi_{\mathsf{pcp}}|_{I_{\mathbb{x},r}}\right) \;\middle|\; \pi_{\mathsf{pcp}} \leftarrow \mathcal{P}_{\mathsf{pcp}}(\mathbb{x}, \mathbb{w}), \quad r \leftarrow \{0,1\}^a \right\}. \tag{2}$$
The two samples on the right are independent. The simulator outputs the coins and answers jointly; the definition does not give a simulator conditional on an arbitrarily prescribed coin string.

Theorem A.26. *Let* $R \in \mathsf{NTIME}[T]$ *with* $T(n) \geq n$*. There is a binary non-adaptive PCP of knowledge for* R *with perfect completeness, constant knowledge threshold at most* $1/2$*, proof length* $\mathsf{poly}(T(n))$*, randomness* $O(\log T(n))$*. Its prover and decoder run in time* $\mathsf{poly}(T(n))$*; its query indices and constant-size decision predicate are computable in time* $\mathsf{poly}(n, \log T(n))$*. The PCP additionally satisfies Definition A.25, with simulator time* $\mathsf{poly}(T(n))$*.*

Proof sketch. We apply the private-predicate composition of Ishai & Weiss (2014, Construction 3 and Lemma 2), specialized to PCPs, to the PCP of knowledge in (Babai et al., 1991) (see Barak & Goldreich (2008, Appendix A)), using the binary exact zero-knowledge PCPP of (Ishai & Weiss, 2014, Corollary 1) for the inner private-predicate relation. □

A.6 VECTOR COMMITMENT SCHEMES

Definition A.27. *A* **vector commitment scheme** *is a tuple of algorithms* $\mathsf{VC} = (\mathsf{VC.Gen}, \mathsf{VC.Commit}, \mathsf{VC.Open}, \mathsf{VC.Check})$ *with the following syntax.*

- $\mathsf{VC.Gen}(1^\lambda, 1^\tau, N, \ell_\mathsf{m})$ *outputs public parameters* pp_vc *specifying a vector length* $N \geq 1$ *and a message space* $\mathcal{M} \subseteq \{0,1\}^{\leq \ell_\mathsf{m}}$.
- $\mathsf{VC.Commit}(\mathsf{pp}_\mathsf{vc}, \mathbf{m})$*, on input* $\mathbf{m} = (m_1, \ldots, m_N) \in \mathcal{M}^N$*, outputs a commitment* c *and private opening state* st.
- $\mathsf{VC.Open}(\mathsf{pp}_\mathsf{vc}, \mathsf{st}, i)$*, on input a position* $i \in [N]$*, outputs an opening* π_vc *for* m_i.
- $\mathsf{VC.Check}(\mathsf{pp}_\mathsf{vc}, c, i, m, \pi_\mathsf{vc})$ *deterministically outputs a bit indicating whether* π_vc *is a valid opening of* c *to* m *at position* i*. It rejects malformed or oversized components, indices outside* $[N]$*, and messages outside* $\mathcal{M}$.

There exists a predicate $\mathsf{Adm}_\mathsf{VC}(1^\lambda, 1^\tau, N, \ell_\mathsf{m}, \mathsf{pp}_\mathsf{vc})$ *that accepts exactly the parameters in the support of* $\mathsf{VC.Gen}(1^\lambda, 1^\tau, N, \ell_\mathsf{m})$*. The scheme has perfect completeness, binding error* β^f_vc *relative to* f*, and strong hiding error* η_vc *if the following hold.*

- **Perfect completeness.** *For every* $\lambda, N, \ell_\mathsf{m}$*, every* $\mathsf{pp}_\mathsf{vc} \in \mathsf{Supp}(\mathsf{VC.Gen}(1^\lambda, 1^\tau, N, \ell_\mathsf{m}))$*, every* $\mathbf{m} \in \mathcal{M}^N$*, and every* $i \in [N]$*,*

$$\Pr\left[\mathsf{VC.Check}(\mathsf{pp}_\mathsf{vc}, c, i, m_i, \pi_\mathsf{vc}) = 1 \;\middle|\; \begin{array}{l}(c, \mathsf{st}) \leftarrow \mathsf{VC.Commit}(\mathsf{pp}_\mathsf{vc}, \mathbf{m}) \\ \pi_\mathsf{vc} \leftarrow \mathsf{VC.Open}(\mathsf{pp}_\mathsf{vc}, \mathsf{st}, i)\end{array}\right] = 1 \ .$$

- **Binding relative to an external oracle.** *For every fixed* $f \in \mathcal{O}_{\lambda,n}$*, every* q*-query* t*-time adversary* $\mathcal{A}$*, and every* N, ℓ_m*,*

$$\Pr\left[\begin{array}{l} m \neq m' \\ \wedge\, \mathsf{VC.Check}(\mathsf{pp}_\mathsf{vc}, c, i, m, \pi_\mathsf{vc}) = 1 \\ \wedge\, \mathsf{VC.Check}(\mathsf{pp}_\mathsf{vc}, c, i, m', \pi'_\mathsf{vc}) = 1 \end{array} \;\middle|\; \begin{array}{l}\mathsf{pp}_\mathsf{vc} \leftarrow \mathsf{VC.Gen}(1^\lambda, 1^\tau, N, \ell_\mathsf{m}) \\ (c, i, m, m', \pi_\mathsf{vc}, \pi'_\mathsf{vc}) \leftarrow \mathcal{A}^f(1^\lambda, \mathsf{pp}_\mathsf{vc})\end{array}\right] \leq \beta^f_\mathsf{vc}(\lambda, N, \ell_\mathsf{m}, q, t) \ .$$

  This property is also called **position binding**. *Binding is computational relative to* $\mathcal{O}$ *if for every* $f \leftarrow \mathcal{O}_{\lambda,n}$*, this bound is negligible in* λ *whenever* $\tau, n, N, \ell_\mathsf{m}, q, t$ *are polynomially bounded in* λ.
- **Strong statistical hiding with error** η_vc**.** *For every admissible* pp_vc*,* $\mathbf{m}^0, \mathbf{m}^1 \in \mathcal{M}^N$ *and* $I \subseteq [N]$ *such that* $\mathbf{m}^0[i] = \mathbf{m}^1[i]$ *for every* $i \in I$*, the statistical distance between* $\mathcal{D}_0$ *and* $\mathcal{D}_1$ *is at most* $\eta_\mathsf{vc}(\lambda, \tau, N, \ell_\mathsf{m})$*, where for* $b \in \{0,1\}$*,* $\mathcal{D}_b$ *is defined by*

$$\mathcal{D}_b \coloneqq \left\{(c, (\mathbf{m}^b[i], \pi^{(i)}_\mathsf{vc})_{i \in I}) \;\middle|\; \begin{array}{l}(c, \mathsf{st}) \leftarrow \mathsf{VC.Commit}(\mathsf{pp}_\mathsf{vc}, \mathbf{m}^b) \\ \forall\, i \in I\colon \pi^{(i)}_\mathsf{vc} \leftarrow \mathsf{VC.Open}(\mathsf{pp}_\mathsf{vc}, \mathsf{st}, i)\end{array}\right\} \ .$$

Definition A.28. *A vector commitment scheme* $\mathsf{VC} = (\mathsf{VC.Gen}, \mathsf{VC.Commit}, \mathsf{VC.Open}, \mathsf{VC.Check})$ *is* **succinct** *if for every security parameter* λ*, vector length* N*, and message-size bound* ℓ_m*, all of the following is bounded by* $\mathsf{poly}(\lambda, \tau, \ell_\mathsf{m}, \log N)$*: the running time of* $\mathsf{VC.Gen}$*,* Adm_VC*,* $\mathsf{VC.Check}$*, the commitment and opening lengths; and* $\mathsf{VC.Commit}$ *and* $\mathsf{VC.Open}$ *run in time polynomial in* $(\lambda, \tau, N, \ell_\mathsf{m})$.

Theorem A.29. *Let* $\mathcal{O}$ *be an oracle. Suppose there exists a keyed hash family* H *that is collision resistant relative to* f *for all* $f \in \mathcal{O}$*. Then there exists a succinct vector commitment scheme satisfying Definition A.27 with perfect completeness, computational position binding relative to* $\mathcal{O}$*, and strong statistical hiding satisfying*

$$\eta_\mathsf{vc}(\lambda, \tau, N, \ell_\mathsf{m}) \leq 2^{-\tau} \tag{3}$$

for every $\lambda, \tau, N, \ell_\mathsf{m} \in \mathbb{N}$.

Proof sketch. We use the statistically hiding commitment of Halevi & Micali (1996), with computational security parameter λ and its statistical-hiding parameter chosen independently so that the hiding error is at most $2^{-\kappa}$, where $\kappa \coloneqq \tau + \lceil \log N \rceil$. Halevi–Micali allow the computational-binding and statistical-hiding parameters to be chosen independently; increasing the latter is achieved by enlarging the auxiliary randomness length without changing the security parameter of the underlying collision-resistant hash family. We commit independently to the N vector entries and place the resulting commitments at the leaves of a binary Merkle tree built from the collision-resistant hash family. An opening consists of the corresponding leaf decommitment and authentication path. Perfect

completeness is immediate. Two accepting openings of the same position to distinct values yield either a double opening of the underlying commitment or a collision in the Merkle tree, proving position binding. For strong statistical hiding, fix two vectors agreeing on the opened positions. Hybridizing over the remaining leaf commitments gives statistical distance at most $N \cdot 2^{-\kappa} \leq 2^{-\tau}$. The Merkle root and authentication paths are deterministic functions of the leaf commitments, so postprocessing cannot increase this distance. Succinctness follows from the underlying commitment and the $O(\log N)$-length authentication paths. □

B USEFUL RELATIONS

Useful relations, formally defined below, have sufficient properties for proving our lower bound. In particular, the oracle can be modified at a few places to insert a witness without affecting the validity of any other witness.

In Section B.1, we establish that there exists a useful relation in the ROM and we explore further properties of useful relations in Section B.2.

Definition B.1. *A relativized relation $R_{\mathcal{O}}$ is a* **useful relation with weight** ω *if for every $\lambda, n \in \mathbb{N}$,*

$$\Pr\left[\, 0^n \notin L(R_f) \mid f \leftarrow \mathcal{O}_{\lambda,n} \,\right] = \omega(\lambda, n) \ ,$$

and there exist

- *a finite non-empty witness domain $\mathcal{W}_{\lambda,n} \subseteq \{0,1\}^{\geq 1}$,*
- *a family of pairwise disjoint query sets $\{Q_{\mathbb{w}}\}_{\mathbb{w} \in \mathcal{W}_{\lambda,n}}$,*
- *a function* Insert *that on input $f \in \mathcal{O}_{\lambda,n}, \mathbb{w} \in \mathcal{W}_{\lambda,n}$ outputs some $f' \in \mathcal{O}_{\lambda,n}$,*

such that

1. *One can sample $\mathbb{w} \leftarrow \mathcal{W}_{\lambda,n}$ in time $O(\log |\mathcal{W}_{\lambda,n}|)$,*
2. *Every candidate witness for 0^n is in $\mathcal{W}_{\lambda,n}$, namely, for every $f \in \mathcal{O}_{\lambda,n}$, $\mathsf{Witness}^{R_{\mathcal{O}}}(f, 0^n) \subseteq \mathcal{W}_{\lambda,n}$,*
3. *There exists an $O(|q|)$-time algorithm that computes the set $\{\mathbb{w} : q \in Q_{\mathbb{w}}\}$ on input q,*
4. *For every $f \in \mathcal{O}_{\lambda,n}$, $\mathbb{w} \in \mathcal{W}_{\lambda,n}$, and $q \notin Q_{\mathbb{w}}$, we have $f(q) = \mathsf{Insert}(f, \mathbb{w})(q)$,*
5. *There exists an $O(|q|)$-time, 1-query oracle algorithm which, given input $(q, \mathbb{w})$ and oracle access to f, outputs $\mathsf{Insert}(f, \mathbb{w})(q)$,*
6. *For every $f \in \mathcal{O}_{\lambda,n}$ and $\mathbb{w} \in \mathcal{W}_{\lambda,n}$, we have $\mathsf{Witness}^{R_{\mathcal{O}}}(\mathsf{Insert}(f, \mathbb{w}), 0^n) = \mathsf{Witness}^{R_{\mathcal{O}}}(f, 0^n) \cup \{\mathbb{w}\}$,*
7. *For every $f_{(0)} \in \mathcal{O}_{\lambda,n}, \mathbb{w} \in \mathcal{W}_{\lambda,n}$ such that $\mathsf{Witness}^{R_{\mathcal{O}}}(f_{(0)}, 0^n) = \{\mathbb{w}\}$,*

$$\left|\{f \in \mathcal{O}_{\lambda,n} : f_{(0)} = \mathsf{Insert}(f, \mathbb{w}) \wedge 0^n \notin L(R_f)\}\right| = |\mathcal{W}_{\lambda,n}| - 1 \ .$$

Query predicates. We now set up notation used throughout the following sections. The main argument in our lower bound rests on a single event: whether an algorithm queries the oracle in the query set $Q_{\mathbb{w}}$.

Definition B.2. *Let A be an interactive oracle algorithm. For every oracle $f \in \mathcal{O}$, input z, randomness r, incoming messages $(a_i)_{i \in [\mathsf{k}]}$, and set S, define*

$$\mathrm{Q}_A(f, z, (a_i)_{i \in [\mathsf{k}]}, r, S) \coloneqq \begin{cases} 1 & \text{if } A^f(z, (a_i)_{i \in [j-1]}; r) \text{ queries } f \text{ at } q \\ & \quad \text{for some } q \in S \text{ and } j \in [\mathsf{k}+1] \\ 0 & \text{otherwise} \end{cases} .$$

If A is a non-interactive algorithm, we set the incoming messages as $\perp$ and write $\mathrm{Q}_A(f, z, \perp, r, S)$.

When $\mathrm{Q}_A(f, z, (a_i)_{i \in [\mathsf{k}]}, r, Q_{\mathbb{w}}) = 1$ for some $\mathbb{w} \in \mathcal{W}_{\lambda,n}$, we say that $A^f(z, (a_i)_{i \in [\mathsf{k}]}; r)$ queries a witness location for $\mathbb{w}$.

We define the predicate specifically for the argument verifier since we repeatedly instantiate the witness-query predicate for the verifier.

Definition B.3. *Let $\mathsf{ARG} = (\mathcal{P}, \mathcal{V})$ be a relativized interactive argument for a useful relation $R_{\mathcal{O}}$. When the security parameter $\lambda \in \mathbb{N}$ and instance size $n \in \mathbb{N}$ are clear from the context, we use the following notation. The* **verifier query predicate** $\mathrm{Q}_{\mathrm{V}}(f, (b_i)_{i \in [\mathsf{k}]}, \rho, \mathbb{w}')$ *evaluates to 1 if and only if the verifier $\mathcal{V}^f(1^\lambda, 0^n, (b_i)_{i \in [j-1]}; \rho)$ queries f at some $q \in Q_{\mathbb{w}'}$ for some $j \in [\mathsf{k}+1]$.*

B.1 A USEFUL RELATION IN THE ROM

We first recall the relativized relation considered in (Barbara et al., 2025).

Definition B.4 ((Barbara et al., 2025, Definition 5.3))**.** *Let $t\colon \mathbb{N} \to \mathbb{N}$ be a time-constructible function with $t(n) \geq n$ for every $n \in \mathbb{N}$ and $\mathcal{U}$ be the random oracle over $((\{0,1\}^{t(n)+\lceil \log t(n)\rceil+1}, \{0,1\}^{\lambda}))_{\lambda,n\in\mathbb{N}}$. Let $e_{k,i}$ denote the $(\lceil \log k\rceil + 1)$-bit string that represents $i \in [k]$. We define $R_{\mathcal{U}} = \{R_u\}_{\lambda,n\in\mathbb{N},u\in\mathcal{U}_{\lambda,n}}$ as follows.*

$$R_u \coloneqq \left\{ (\mathtt{x},\mathtt{w}) \in \{0,1\}^n \times \{0,1\}^{t(n)} \;\middle|\; \begin{array}{l} \mathtt{x} = 0^n \\ \wedge \forall\, i \in [t(n)], u(\mathtt{w}\|e_{t(n),i})_1 = 0 \end{array} \right\} .$$

Lemma B.5. *Let $t\colon \mathbb{N} \to \mathbb{N}$ be a time-constructible function with $t(n) \geq n$ for every $n \in \mathbb{N}$ and $\mathcal{U}$ be the random oracle over $((\{0,1\}^{t(n)+\lceil \log t(n)\rceil+1}, \{0,1\}^{\lambda}))_{\lambda,n\in\mathbb{N}}$. The relativized relation $R_{\mathcal{U}}$ defined in Definition B.4 is a useful relation with weight $\omega(\lambda,n) = (1-2^{-t(n)})^{2^{t(n)}}$, witness domain $\mathcal{W}_{\lambda,n} = \{0,1\}^{t(n)}$, query set $Q_{\mathtt{w}} = \{\mathtt{w}\|e_{t(n),i} : i \in [t(n)]\}$, and the function* Insert *defined as follows.*

$\mathsf{Insert}(u,\mathtt{w})$*:*
1. *Initialize $u' \coloneqq u$.*
2. *For all $i \in [t(n)]$: $u'(\mathtt{w}\|e_{t(n),i})_1 \coloneqq 0$.*
3. *Output u'.*

Furthermore,

$$R_{\mathcal{U}} \in \mathsf{NTIME}^{\mathcal{U}}[t] .$$

Remark B.6. For the NTIME relation $R_{\mathcal{U}}$ in Definition B.4, for an interactive algorithm A, input z, messages $(a_i)_{i\in[\mathsf{k}]}$, randomness r, oracle u, $n \in \mathbb{N}$, and witness $\mathtt{w} \in \{0,1\}^{t(n)}$,

$$\mathrm{Q}_A(u,z,(a_i)_{i\in[\mathsf{k}]},r,Q_{\mathtt{w}}) = \begin{cases} 1 & \text{if } A^u(z,(a_i)_{i\in[j-1]};r) \text{ queries } u \text{ at } \mathtt{w}\|e_{t(n),i} \\ & \quad \text{for some } i \in [t(n)], j \in [\mathsf{k}+1] \\ 0 & \text{otherwise} . \end{cases} ,$$

In the remainder of this section, we prove Lemma B.5.

Proof of Lemma B.5.

Claim B.7 ((Barbara et al., 2025, Lemma 5.4))**.**

$$R_{\mathcal{U}} \in \mathsf{NTIME}^{\mathcal{U}}[t] .$$

We fix $\lambda, n \in \mathbb{N}$.

Claim B.8.

$$\Pr\left[\, 0^n \notin L(R_u) \mid u \leftarrow \mathcal{U}_{\lambda,n} \,\right] = \left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}} .$$

Proof. We have

$$\Pr\left[\, 0^n \notin L(R_u) \mid u \leftarrow \mathcal{U}_{\lambda,n} \,\right] = \Pr\left[\, \forall\, \mathtt{w} \in \mathcal{W}_{\lambda,n}, (0^n,\mathtt{w}) \notin R_u \mid u \leftarrow \mathcal{U}_{\lambda,n} \,\right]$$

(by Definition A.4)

$$= \prod_{\mathtt{w}\in\mathcal{W}_{\lambda,n}} \Pr\left[\, (0^n,\mathtt{w}) \notin R_u \mid u \leftarrow \mathcal{U}_{\lambda,n} \,\right]$$

(by Definition B.4, the events $\{(0^n,\mathtt{w}) \in R_u\}_{\mathtt{w}\in\mathcal{W}_{\lambda,n}}$ are mutually independent)

$$= \prod_{\mathtt{w}\in\mathcal{W}_{\lambda,n}} \left(1 - \frac{1}{2^{t(n)}}\right)$$

($(0^n,\mathtt{w}) \in R_u$ iff for every $i \in [t(n)]$, $u(\mathtt{w}\|e_{t(n),i})_1 = 0$)

$$= \left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}} \quad .$$

□

For every $u \in \mathcal{U}_{\lambda,n}$, by the definition of $Q_{\mathbb{w}}$, the sets $Q_{\mathbb{w}_0}$ and $Q_{\mathbb{w}_1}$ are disjoint if $\mathbb{w}_0 \neq \mathbb{w}_1$. Furthermore, given an oracle query $q \in \{0,1\}^{t(n)+\lceil \log t(n) \rceil + 1}$, one can compute the set $\{\mathbb{w} : q \in Q_{\mathbb{w}}\}$ in time $O(t(n))$ with no queries to the oracle as follows.

1. Parse q as $\mathbb{w} \| e_{t(n),i}$.
2. If $i \in [t(n)]$, output $\{\mathbb{w}\}$. Otherwise, output $\emptyset$.

For every query q, one can compute $\mathsf{Insert}(u, \mathbb{w})(q)$ in time $O(t(n))$ with 1 query to u as follows.

$A^u(q, \mathbb{w})$:
1. Query the oracle at q: $\mathsf{a} \coloneqq u(q)$.
2. Parse q as $(\mathbb{w}', e_{t(n),i})$.
3. If $\mathbb{w}' = \mathbb{w}$ and $i \in [t(n)]$, set $\mathsf{a}_1 \coloneqq 0$.
4. Output a.

Let $u' \coloneqq \mathsf{Insert}(u, \mathbb{w})$. We have $u' \in \mathcal{U}_{\lambda,n}$ by the definition of $\mathcal{U}_{\lambda,n}$. We now argue

$$\mathsf{Witness}^{R_{\mathcal{U}}}(\mathsf{Insert}(u, \mathbb{w}), 0^n) = \mathsf{Witness}^{R_{\mathcal{U}}}(u, 0^n) \cup \{\mathbb{w}\} \quad .$$

By Definition B.4, $(0^n, \mathbb{w}) \in R_{\mathsf{Insert}(u,\mathbb{w})}$. For every $u \in \mathcal{U}_{\lambda,n}$ and $\mathbb{w} \in \{0,1\}^{t(n)}$, by the definitions of R_u and $Q_{\mathbb{w}}$, whether $(0^n, \mathbb{w}) \in R_u$ is a function of $\{u(q) : q \in Q_{\mathbb{w}}\}$, i.e. the oracle's answers to the queries in $Q_{\mathbb{w}}$. Since the sets $Q_{\mathbb{w}}$ and $Q_{\mathbb{w}'}$ are disjoint for every $\mathbb{w} \neq \mathbb{w}'$, we conclude that inserting $\mathbb{w}$ does not change whether any $\mathbb{w}' \neq \mathbb{w}$ is a witness.

We define a function $\mathsf{Recover}$ that works as follows to prove the remaining properties. For $u \in \mathcal{U}_{\lambda,n}$, $\mathbb{w} \in \{0,1\}^{t(n)}$, $s \in \{0,1\}^{t(n)}$,

$\mathsf{Recover}(u, \mathbb{w}, s)$:
1. Initialize $u' \coloneqq u$.
2. For all $i \in [t(n)]$: $u'(\mathbb{w} \| e_{t(n),i})_1 \coloneqq s_i$.
3. Output u'.

Claim B.9. *For every $u_{(0)} \in \mathcal{U}_{\lambda,n}, \mathbb{w} \in \{0,1\}^{t(n)}$ such that $\mathsf{Witness}^{R_{\mathcal{U}}}(u_{(0)}, 0^n) = \{\mathbb{w}\}$,*

$$2^{t(n)} - 1 = \left|\{u \in \mathcal{U}_{\lambda,n} : u_{(0)} = \mathsf{Insert}(u, \mathbb{w}) \wedge 0^n \notin L(R_u)\}\right| \quad .$$

Proof. We have $|\mathcal{W}_{\lambda,n}| = 2^{t(n)}$ and

$$\begin{aligned}
&\left|\{u \in \mathcal{U}_{\lambda,n} : u_{(0)} = \mathsf{Insert}(u, \mathbb{w}) \wedge 0^n \notin L(R_u)\}\right| \\
&= \left|\{\mathsf{Recover}(u_{(0)}, \mathbb{w}, s) : s \in \{0,1\}^{t(n)} \setminus \{0^{t(n)}\}\}\right| \\
&= 2^{t(n)} - 1 \quad .
\end{aligned}$$

□

□

B.2 Properties of useful relations

We fix $\lambda, n \in \mathbb{N}$ throughout this section.

Claim B.10. *Let $\mathcal{O}$ be a uniform oracle and $R_{\mathcal{O}}$ be a useful relation. For every $\mathbb{w}_0^\star \in \mathcal{W}_{\lambda,n}$, $f_{(0)}^\star \in \mathcal{O}_{\lambda,n}$ such that $\mathsf{Witness}^{R_{\mathcal{O}}}(f_{(0)}^\star, 0^n) = \{\mathbb{w}_0^\star\}$,*

$$\Pr\left[\begin{array}{l|l} f_{(0)} = f_{(0)}^\star & f \leftarrow \mathcal{O}_{\lambda,n} \\ \wedge\, \mathbb{w}_0 = \mathbb{w}_0^\star & \mathbb{w}_0 \leftarrow \mathcal{W}_{\lambda,n} \\ \wedge\, 0^n \notin L(R_f) & f_{(0)} \coloneqq \mathsf{Insert}(f, \mathbb{w}_0) \end{array}\right]$$

$$= \left(1 - \frac{1}{|\mathcal{W}_{\lambda,n}|}\right) \cdot \Pr\left[\begin{array}{l|ll} f = f^\star_{(0)} & & f \leftarrow \mathcal{O}_{\lambda,n} \\ \wedge\, \mathrm{w}_0 = \mathrm{w}^\star_0 & & \mathrm{w}_0 \leftarrow \mathcal{A}^f \\ \wedge\, (0^n, \mathrm{w}_0) \in R_f & & \end{array}\right] ,$$

where $\mathcal{A}$ is defined as follows.

$\mathcal{A}^f$:

1. *Compute* $\mathsf{Witness}^{R_\mathcal{O}}(f, 0^n)$ *by querying* f *as needed.*
2. *If there exists some* w *such that* $\mathsf{Witness}^{R_\mathcal{O}}(f, 0^n) = \{\mathrm{w}\}$*, output* w*. Otherwise, output* $\perp$*.*

Proof. Let

$$S \coloneqq \{f^\star \in \mathcal{O}_{\lambda,n} \colon f^\star_{(0)} = \mathsf{Insert}(f^\star, \mathrm{w}^\star_0) \wedge 0^n \notin L(R_{f^\star})\} .$$

We have

$$\begin{aligned}
&\Pr\left[\begin{array}{l|ll} f_{(0)} = f^\star_{(0)} \wedge \mathrm{w}_0 = \mathrm{w}^\star_0 & & f \leftarrow \mathcal{O}_{\lambda,n} \\ \wedge\, 0^n \notin L(R_f) & & \mathrm{w}_0 \leftarrow \mathcal{W}_{\lambda,n} \\ & & f_{(0)} \coloneqq \mathsf{Insert}(f, \mathrm{w}_0) \end{array}\right] \\
&= \sum_{f^\star \in S} \Pr\left[\begin{array}{l|ll} f = f^\star \wedge \mathrm{w}_0 = \mathrm{w}^\star_0 & & f \leftarrow \mathcal{O}_{\lambda,n} \\ \wedge\, 0^n \notin L(R_f) & & \mathrm{w}_0 \leftarrow \mathcal{W}_{\lambda,n} \end{array}\right] && \text{(by the law of total probability)} \\
&= \sum_{f^\star \in S} \Pr\left[f = f^\star \wedge \mathrm{w}_0 = \mathrm{w}^\star_0 \,\middle|\, \begin{array}{l} f \leftarrow \mathcal{O}_{\lambda,n} \\ \mathrm{w}_0 \leftarrow \mathcal{W}_{\lambda,n} \end{array}\right] && \text{(by the definition of } S\text{)} \\
&= \sum_{f^\star \in S} \Pr\left[f = f^\star_{(0)} \wedge \mathrm{w}_0 = \mathrm{w}^\star_0 \,\middle|\, \begin{array}{l} f \leftarrow \mathcal{O}_{\lambda,n} \\ \mathrm{w}_0 \leftarrow \mathcal{W}_{\lambda,n} \end{array}\right] && \text{(by uniformity of } \mathcal{O}\text{)} \\
&= \frac{1}{|\mathcal{W}_{\lambda,n}|} \cdot \sum_{f^\star \in S} \Pr\left[f = f^\star_{(0)} \,\middle|\, f \leftarrow \mathcal{O}_{\lambda,n}\right] && \text{(by how } \mathrm{w}_0 \text{ is sampled)} \\
&= \frac{|S|}{|\mathcal{W}_{\lambda,n}|} \cdot \Pr\left[f = f^\star_{(0)} \,\middle|\, f \leftarrow \mathcal{O}_{\lambda,n}\right] && \text{(by summing over } f^\star\text{)} \\
&= \frac{|S|}{|\mathcal{W}_{\lambda,n}|} \cdot \frac{\Pr\left[f = f^\star_{(0)} \wedge \mathrm{w}_0 = \mathrm{w}^\star_0 \,\middle|\, \begin{array}{l} f \leftarrow \mathcal{O}_{\lambda,n} \\ \mathrm{w}_0 \leftarrow \mathcal{A}^f \end{array}\right]}{\Pr\left[\mathrm{w}_0 = \mathrm{w}^\star_0 \,\middle|\, \mathrm{w}_0 \leftarrow \mathcal{A}^{f^\star_{(0)}}\right]} && \text{(by the chain rule)} \\
&= \frac{|S|}{|\mathcal{W}_{\lambda,n}|} \cdot \Pr\left[\begin{array}{l|ll} f = f^\star_{(0)} \wedge \mathrm{w}_0 = \mathrm{w}^\star_0 & & f \leftarrow \mathcal{O}_{\lambda,n} \\ \wedge\, (0^n, \mathrm{w}_0) \in R_f & & \mathrm{w}_0 \leftarrow \mathcal{A}^f \end{array}\right] && \text{(since } \mathsf{Witness}^{R_\mathcal{O}}(f^\star_{(0)}, 0^n) = \{\mathrm{w}^\star_0\}\text{)} \\
&= \left(1 - \frac{1}{|\mathcal{W}_{\lambda,n}|}\right) \cdot \Pr\left[\begin{array}{l|ll} f = f^\star_{(0)} \wedge \mathrm{w}_0 = \mathrm{w}^\star_0 & & f \leftarrow \mathcal{O}_{\lambda,n} \\ \wedge\, (0^n, \mathrm{w}_0) \in R_f & & \mathrm{w}_0 \leftarrow \mathcal{A}^f \end{array}\right] . \\
& && \text{(since } R_\mathcal{O} \text{ is a useful relation)}
\end{aligned}$$

□

Lemma B.11. *Let $\mathcal{O}$ be a uniform oracle and $R_\mathcal{O}$ be a useful relation. For every set Z, predicate ϕ over $\mathcal{O}_{\lambda,n} \times (\mathcal{W}_{\lambda,n} \cup \{\perp\}) \times Z$, and family of distributions $\{\mathcal{D}_{(f_{(0)},\mathrm{w}_0)}\}_{(f_{(0)},\mathrm{w}_0) \in \mathcal{O}_{\lambda,n} \times (\mathcal{W}_{\lambda,n} \cup \{\perp\})}$ over Z,*

$$\begin{aligned}
&\Pr\left[\begin{array}{l|ll} \phi(f_{(0)}, \mathrm{w}_0, z) = 1 & & f \leftarrow \mathcal{O}_{\lambda,n} \\ \wedge\, 0^n \notin L(R_f) & & \mathrm{w}_0 \leftarrow \mathcal{W}_{\lambda,n} \\ & & f_{(0)} \coloneqq \mathsf{Insert}(f, \mathrm{w}_0) \\ & & z \leftarrow \mathcal{D}_{(f_{(0)},\mathrm{w}_0)} \end{array}\right] \\
&= \left(1 - \frac{1}{|\mathcal{W}_{\lambda,n}|}\right) \cdot \Pr\left[\begin{array}{l|ll} \phi(f, \mathrm{w}_0, z) = 1 & & f \leftarrow \mathcal{O}_{\lambda,n} \\ \wedge\, (0^n, \mathrm{w}_0) \in R_f & & \mathrm{w}_0 \leftarrow \mathcal{A}^f \\ & & z \leftarrow \mathcal{D}_{(f,\mathrm{w}_0)} \end{array}\right] ,
\end{aligned}$$

where $\mathcal{A}$ is defined as in Claim B.10.

Proof. Let

$$\mathrm{EXP} := \left\{ \begin{array}{l} f \leftarrow \mathcal{O}_{\lambda,n} \\ \mathbb{w}_0 \leftarrow \mathcal{W}_{\lambda,n} \\ f_{(0)} := \mathsf{Insert}(f, \mathbb{w}_0) \\ z \leftarrow \mathcal{D}_{(f_{(0)}, \mathbb{w}_0)} \end{array} \right\} \quad \text{and} \quad \mathrm{EXP}' := \left\{ \begin{array}{l} f \leftarrow \mathcal{O}_{\lambda,n} \\ \mathbb{w}_0 \leftarrow \mathcal{A}^f \\ z \leftarrow \mathcal{D}_{(f, \mathbb{w}_0)} \end{array} \right\} .$$

We claim that

$$\Pr_{\mathrm{EXP}} \left[\begin{array}{l} \phi(f_{(0)}, \mathbb{w}_0, z) = 1 \wedge 0^n \notin L(R_f) \\ \wedge\, \mathsf{Witness}^{R_{\mathcal{O}}}(f_{(0)}, 0^n) = \{\mathbb{w}_0\} \end{array} \right] = \left(1 - \frac{1}{|\mathcal{W}_{\lambda,n}|}\right) \cdot \Pr_{\mathrm{EXP}'} \left[\begin{array}{l} \phi(f, \mathbb{w}_0, z) = 1 \\ \wedge\, (0^n, \mathbb{w}_0) \in R_f \end{array} \right] . \tag{4}$$

The lemma follows from Eq. 4 because

$$\begin{aligned} \left(1 - \frac{1}{|\mathcal{W}_{\lambda,n}|}\right) \cdot \Pr_{\mathrm{EXP}'} \left[\begin{array}{l} \phi(f, \mathbb{w}_0, z) = 1 \\ \wedge\, (0^n, \mathbb{w}_0) \in R_f \end{array} \right] &= \Pr_{\mathrm{EXP}} \left[\begin{array}{l} \phi(f_{(0)}, \mathbb{w}_0, z) = 1 \wedge 0^n \notin L(R_f) \\ \wedge\, \mathsf{Witness}^{R_{\mathcal{O}}}(f_{(0)}, 0^n) = \{\mathbb{w}_0\} \end{array} \right] \\ &\qquad \text{(by Eq. 4)} \\ &= \Pr_{\mathrm{EXP}} \left[\begin{array}{l} \phi(f_{(0)}, \mathbb{w}_0, z) = 1 \\ \wedge\, 0^n \notin L(R_f) \end{array} \right] \\ &\quad - \Pr_{\mathrm{EXP}} \left[\begin{array}{l} \phi(f_{(0)}, \mathbb{w}_0, z) = 1 \wedge 0^n \notin L(R_f) \\ \wedge\, \mathsf{Witness}^{R_{\mathcal{O}}}(f_{(0)}, 0^n) \neq \{\mathbb{w}_0\} \end{array} \right] \\ &\qquad \text{(by the law of total probability)} \\ &= \Pr_{\mathrm{EXP}} \left[\begin{array}{l} \phi(f_{(0)}, \mathbb{w}_0, z) = 1 \\ \wedge\, 0^n \notin L(R_f) \end{array} \right] . \\ &\qquad \text{(since } \mathsf{Insert} \text{ adds exactly } \mathbb{w} \text{ to the witness set by Definition B.1)} \end{aligned}$$

We now show that Eq. 4 holds:

$$\begin{aligned} &\Pr_{\mathrm{EXP}} \left[\; \phi(f_{(0)}, \mathbb{w}_0, z) = 1 \wedge 0^n \notin L(R_f) \wedge \mathsf{Witness}^{R_{\mathcal{O}}}(f_{(0)}, 0^n) = \{\mathbb{w}_0\} \; \right] \\ &= \sum_{\substack{f^\star_{(0)} \in \mathcal{O}_{\lambda,n}, \mathbb{w}^\star_0 \in \mathcal{W}_{\lambda,n} \\ \text{s.t. } \mathsf{Witness}^{R_{\mathcal{O}}}(f^\star_{(0)}, 0^n) = \{\mathbb{w}^\star_0\}}} \Pr \left[\begin{array}{l|l} f_{(0)} = f^\star_{(0)} & f \leftarrow \mathcal{O}_{\lambda,n} \\ \wedge\, \mathbb{w}_0 = \mathbb{w}^\star_0 & \mathbb{w}_0 \leftarrow \mathcal{W}_{\lambda,n} \\ \wedge\, 0^n \notin L(R_f) & f_{(0)} := \mathsf{Insert}(f, \mathbb{w}_0) \end{array} \right] \\ &\quad \cdot \Pr \left[\phi(f^\star_{(0)}, \mathbb{w}^\star_0, z) = 1 \;\middle|\; z \leftarrow \mathcal{D}_{(f^\star_{(0)}, \mathbb{w}^\star_0)} \right] \quad \text{(by the law of total probability and chain rule)} \\ &= \sum_{\substack{f^\star_{(0)} \in \mathcal{O}_{\lambda,n}, \mathbb{w}^\star_0 \in \mathcal{W}_{\lambda,n} \\ \text{s.t. } \mathsf{Witness}^{R_{\mathcal{O}}}(f^\star_{(0)}, 0^n) = \{\mathbb{w}^\star_0\}}} \left(1 - \frac{1}{|\mathcal{W}_{\lambda,n}|}\right) \cdot \Pr \left[\begin{array}{l|l} f = f^\star_{(0)} & f \leftarrow \mathcal{O}_{\lambda,n} \\ \wedge\, \mathbb{w}_0 = \mathbb{w}^\star_0 & \mathbb{w}_0 \leftarrow \mathcal{A}^f \\ \wedge\, (0^n, \mathbb{w}_0) \in R_f & \end{array} \right] \\ &\quad \cdot \Pr \left[\phi(f^\star_{(0)}, \mathbb{w}^\star_0, z) = 1 \;\middle|\; z \leftarrow \mathcal{D}_{(f^\star_{(0)}, \mathbb{w}^\star_0)} \right] \quad \text{(by Claim B.10)} \\ &= \left(1 - \frac{1}{|\mathcal{W}_{\lambda,n}|}\right) \cdot \Pr_{\mathrm{EXP}'} \left[\; \phi(f, \mathbb{w}_0, z) = 1 \wedge (0^n, \mathbb{w}_0) \in R_f \; \right] . \\ &\qquad \text{(by the chain rule and law of total probability, which applies here since } (0^n, \perp) \notin R_f) \end{aligned}$$

□

C TECHNICAL LEMMAS

This section proves the lemma at the heart of the lower bound: a complete and sound verifier must query a witness location with constant probability. Section C.1 shows that a change to the oracle goes undetected unless the changed location is queried, and Section C.2 derives the lemma from this.

C.1 MODIFYING THE ORACLE TO INSERT A WITNESS

We explore how the queries and output of two interacting oracle algorithms change if we modify the oracles at a few locations. Intuitively, if an oracle algorithm is run on the same input with access to

different oracles, the outcome is the same unless the algorithm queries the oracles at a location at which they differ. The following claim formalizes this idea.

Claim C.1. *Let A_0 and A_1 be interactive oracle algorithms, interacting for k rounds, with respective input domains Z_0 and Z_1. For functions $f_0, f_1 \in \mathsf{Func}(X,Y)$, let $\Delta(f_0,f_1)$ be the set $\{q \in X\colon f_0(q) \neq f_1(q)\}$. For every distribution $\mathcal{D}$ over $\mathsf{Func}(X,Y) \times \mathsf{Func}(X,Y) \times \mathsf{Func}(X,Y) \times \mathsf{Func}(X,Y) \times Z_0 \times Z_1$ and sets Q_0, Q_1, if $\Delta(f_0,f_0') \subseteq Q_0$ and $\Delta(f_1,f_1') \subseteq Q_1$, then on the event*

$$\mathrm{Q}_{A_0}(f_0, z_0, (a_i)_{i\in[\mathsf{k}]}, r_0, Q_0) = 0 \ \wedge\ \mathrm{Q}_{A_1}(f_1, z_1, (b_i)_{i\in[\mathsf{k}]}, r_1, Q_1) = 0 \ , \tag{5}$$

we have

$$\mathsf{View}_{A_0}(f_0, z_0, (a_i)_{i\in[\mathsf{k}]}, r_0) = \mathsf{View}_{A_0}(f_0', z_0, (a_i')_{i\in[\mathsf{k}]}, r_0) \ \text{ and}$$
$$\mathsf{View}_{A_1}(f_1, z_1, (b_i)_{i\in[\mathsf{k}]}, r_1) = \mathsf{View}_{A_1}(f_1', z_1, (b_i')_{i\in[\mathsf{k}]}, r_1) \ ,$$

where the random variables are defined by the following random experiment.

$$\mathrm{EXP} \coloneqq \left\{ \begin{array}{l} (f_0, f_0', f_1, f_1', z_0, z_1) \leftarrow \mathcal{D} \\ \mathsf{d} \xleftarrow{r_0, r_1, (b_1, a_1, \ldots, b_{\mathsf{k}}, a_{\mathsf{k}})} \langle A_0^{f_0}(z_0), A_1^{f_1}(z_1)\rangle \\ \mathsf{d} \xleftarrow{(b_1', a_1', \ldots, b_{\mathsf{k}}', a_{\mathsf{k}}')} \langle A_0^{f_0'}(z_0; r_0), A_1^{f_1'}(z_1; r_1)\rangle \end{array} \right\} .$$

Proof. Suppose Equation 5 holds. In other words, suppose A_0 does not query the oracle in the difference $\Delta(f_0,f_0')$ and A_1 does not query the oracle in the difference $\Delta(f_1,f_1')$ in EXP.

We prove by strong induction on i that for every $i \in [\mathsf{k}]$, $b_i = b_i'$, $a_i = a_i'$,

$$\mathsf{View}_{A_0}(f_0, z_0, (a_j)_{j\in[i-1]}, r_0) = \mathsf{View}_{A_0}(f_0', z_0, (a_j')_{j\in[i-1]}, r_0) \ \text{ and}$$
$$\mathsf{View}_{A_1}(f_1, z_1, (b_j)_{j\in[i]}, r_1) = \mathsf{View}_{A_1}(f_1', z_1, (b_j')_{j\in[i]}, r_1) \ .$$

We begin with proving the base case $i = 1$. Since $f_0(q) = f_0'(q)$ for every $q \notin \Delta(f_0,f_0')$ and A_0 does not query the oracle in $\Delta(f_0,f_0')$, the views $\mathsf{View}_{A_0}(f_0, z_0, \perp, r_0)$ and $\mathsf{View}_{A_0}(f_0', z_0, \perp, r_0)$ are identical. Hence, $b_1 = b_1'$ by Definition A.11.

Since $f_1(q) = f_1'(q)$ for every $q \notin \Delta(f_1,f_1')$ and A_1 does not query the oracle in $\Delta(f_1,f_1')$, the views $\mathsf{View}_{A_1}(f_1, z_1, (b_1), r_1)$ and $\mathsf{View}_{A_1}(f_1', z_1, (b_1), r_1)$ are identical. Using $b_1 = b_1'$, we conclude that the views $\mathsf{View}_{A_1}(f_1, z_1, (b_1), r_1)$ and $\mathsf{View}_{A_1}(f_1', z_1, (b_1'), r_1)$ are identical. Moreover, we have $a_1 = a_1'$ by Definition A.11. This finishes the proof of the base case.

We now assume that for some $i \in [\mathsf{k}]$, for all $j < i$, $b_j = b_j'$, $a_j = a_j'$, $\mathsf{View}_{A_0}(f_0, z_0, (a_k)_{k\in[j-1]}, r_0) = \mathsf{View}_{A_0}(f_0', z_0, (a_k')_{k\in[j-1]}, r_0)$, and $\mathsf{View}_{A_1}(f_1, z_1, (b_k)_{k\in[j]}, r_1) = \mathsf{View}_{A_1}(f_1', z_1, (b_k')_{k\in[j]}, r_1)$. Because A_0 does not query the oracle in $\Delta(f_0,f_0')$, we conclude by the induction hypothesis that the views $\mathsf{View}_{A_0}(f_0, z_0, (a_j)_{j\in[i-1]}, r_0)$ and $\mathsf{View}_{A_0}(f_0', z_0, (a_j')_{j\in[i-1]}, r_0)$ are identical. Therefore, we have $b_i = b_i'$ by Definition A.11. Equation 5, $b_i = b_i'$, and the induction hypothesis together imply that the views $\mathsf{View}_{A_1}(f_1, z_1, (b_j)_{j\in[i]}, r_1)$ and $\mathsf{View}_{A_1}(f_1', z_1, (b_j')_{j\in[i]}, r_1)$ are identical, and, consequently, $a_i = a_i'$ by Definition A.11. Therefore, the claim holds for i, concluding the induction. □

The following claim extends Claim C.1 to evaluate predicates over the view given different oracles.

Claim C.2. *Let A_0 and A_1 be interactive oracle algorithms, interacting for k rounds, with respective input domains Z_0 and Z_1. Let Z be a set and ϕ_1 and ϕ_2 be a predicate over A_1's view and $z \in Z$, respectively.*

For every distribution $\mathcal{D}$ over $\mathsf{Func}(X,Y) \times \mathsf{Func}(X,Y) \times \mathsf{Func}(X,Y) \times \mathsf{Func}(X,Y) \times Z_0 \times Z_1 \times Z \times 2^X \times 2^X$, if $\Delta(f_0,f_0') \subseteq Q_0$ and $\Delta(f_1,f_1') \subseteq Q_1$ with probability 1, then we have

$$\left| \Pr\left[\begin{array}{l} \phi_1(\mathsf{View}_{A_1}(f_1, z_1, (b_i)_{i\in[\mathsf{k}]}, r_1)) = 1 \\ \wedge\, \phi_2(z) = 1 \end{array} \right] - \Pr\left[\begin{array}{l} \phi_1(\mathsf{View}_{A_1}(f_1', z_1, (b_i')_{i\in[\mathsf{k}]}, r_1)) = 1 \\ \wedge\, \phi_2(z) = 1 \end{array} \right] \right|$$
$$\leq \Pr\left[\begin{array}{l} (\mathrm{Q}_{A_0}(f_0, z_0, (a_i)_{i\in[\mathsf{k}]}, r_0, Q_0) = 1 \ \vee\ \mathrm{Q}_{A_1}(f_1, z_1, (b_i)_{i\in[\mathsf{k}]}, r_1, Q_1) = 1) \\ \wedge\, \phi_2(z) = 1 \end{array} \right] ,$$

where the random variables are sampled according to the following random experiment.

$$\mathrm{EXP} \coloneqq \left\{ \begin{array}{l} (f_0, f_0', f_1, f_1', z_0, z_1, z, Q_0, Q_1) \leftarrow \mathcal{D} \\ \mathsf{d} \xleftarrow{r_0, r_1, (b_1, a_1, \ldots, b_\mathsf{k}, a_\mathsf{k})} \langle A_0^{f_0}(z_0), A_1^{f_1}(z_1) \rangle \\ \mathsf{d} \xleftarrow{(b_1', a_1', \ldots, b_\mathsf{k}', a_\mathsf{k}')} \langle A_0^{f_0'}(z_0; r_0), A_1^{f_1'}(z_1; r_1) \rangle \end{array} \right\} .$$

Proof. Let $E \coloneqq \{\mathrm{Q}_{A_0}(f_0, z_0, (a_i)_{i\in[\mathsf{k}]}, r_0, Q_0) = 0 \wedge \mathrm{Q}_{A_1}(f_1, z_1, (b_i)_{i\in[\mathsf{k}]}, r_1, Q_1) = 0\}$ and $E^* \coloneqq \{\phi_2(z) = 1\}$. We have

$$\begin{aligned}
&\Pr\left[\begin{array}{l} \phi_1(\mathsf{View}_{A_1}(f_1, z_1, (b_i)_{i\in[\mathsf{k}]}, r_1)) = 1 \\ \wedge E^* \end{array} \right] - \Pr\left[\phi_1(\mathsf{View}_{A_1}(f_1', z_1, (b_i')_{i\in[\mathsf{k}]}, r_1)) = 1 \wedge E^* \right] \\
&= \Pr\left[\begin{array}{l} \phi_1(\mathsf{View}_{A_1}(f_1, z_1, (b_i)_{i\in[\mathsf{k}]}, r_1)) = 1 \\ \wedge E \wedge E^* \end{array} \right] + \Pr\left[\begin{array}{l} \phi_1(\mathsf{View}_{A_1}(f_1, z_1, (b_i)_{i\in[\mathsf{k}]}, r_1)) = 1 \\ \wedge \neg E \wedge E^* \end{array} \right] \\
&\quad - \Pr\left[\begin{array}{l} \phi_1(\mathsf{View}_{A_1}(f_1', z_1, (b_i')_{i\in[\mathsf{k}]}, r_1)) = 1 \\ \wedge E \wedge E^* \end{array} \right] - \Pr\left[\begin{array}{l} \phi_1(\mathsf{View}_{A_1}(f_1', z_1, (b_i')_{i\in[\mathsf{k}]}, r_1)) = 1 \\ \wedge \neg E \wedge E^* \end{array} \right] \\
&\qquad\qquad \text{(by the law of total probability)} \\
&= \Pr\left[\begin{array}{l} \phi_1(\mathsf{View}_{A_1}(f_1, z_1, (b_i)_{i\in[\mathsf{k}]}, r_1)) = 1 \\ \wedge \neg E \wedge E^* \end{array} \right] - \Pr\left[\begin{array}{l} \phi_1(\mathsf{View}_{A_1}(f_1', z_1, (b_i')_{i\in[\mathsf{k}]}, r_1)) = 1 \\ \wedge \neg E \wedge E^* \end{array} \right] \\
&\qquad \text{(if } E \text{ happens, then } \mathsf{View}_{A_1}(f_1', z_1, (b_i')_{i\in[\mathsf{k}]}, r_1) = \mathsf{View}_{A_1}(f_1, z_1, (b_i)_{i\in[\mathsf{k}]}, r_1) \text{ by Claim C.1)}
\end{aligned}$$

Therefore, by the properties of absolute value and probability,

$$\begin{aligned}
&\left| \Pr\left[\begin{array}{l} \phi_1(\mathsf{View}_{A_1}(f_1, z_1, (b_i)_{i\in[\mathsf{k}]}, r_1)) = 1 \\ \wedge E^* \end{array} \right] - \Pr\left[\begin{array}{l} \phi_1(\mathsf{View}_{A_1}(f_1', z_1, (b_i')_{i\in[\mathsf{k}]}, r_1)) = 1 \\ \wedge E^* \end{array} \right] \right| \\
&= \left| \Pr\left[\begin{array}{l} \phi_1(\mathsf{View}_{A_1}(f_1, z_1, (b_i)_{i\in[\mathsf{k}]}, r_1)) \\ = 1 \wedge \neg E \wedge E^* \end{array} \right] - \Pr\left[\begin{array}{l} \phi_1(\mathsf{View}_{A_1}(f_1', z_1, (b_i')_{i\in[\mathsf{k}]}, r_1)) = 1 \\ \wedge \neg E \wedge E^* \end{array} \right] \right| \\
&\leq \max\left\{ \Pr\left[\begin{array}{l} \phi_1(\mathsf{View}_{A_1}(f_1, z_1, (b_i)_{i\in[\mathsf{k}]}, r_1)) \\ = 1 \wedge \neg E \wedge E^* \end{array} \right], \Pr\left[\begin{array}{l} \phi_1(\mathsf{View}_{A_1}(f_1', z_1, (b_i')_{i\in[\mathsf{k}]}, r_1)) \\ = 1 \wedge \neg E \wedge E^* \end{array} \right] \right\} \\
&\leq \Pr\left[\neg E \wedge E^* \right] \\
&= \Pr\left[\begin{array}{l} (\mathrm{Q}_{A_0}(f_0, z_0, (a_i)_{i\in[\mathsf{k}]}, r_0, Q_0) = 1 \vee \mathrm{Q}_{A_1}(f_1, z_1, (b_i)_{i\in[\mathsf{k}]}, r_1, Q_1) = 1) \\ \wedge \phi_2(z) = 1 \end{array} \right] .
\end{aligned}$$

□

C.2 The verifier should query a witness location

The completeness and soundness of a relativized argument together require the verifier to query a witness location for the prover's witness if the relativized relation is useful as in Section B. In particular, we obtain a lower bound on the probability that the verifier queries a witness location for the prover's witness using the completeness and soundness errors. This bound, formally established in Lemma C.5, serves as the first block of the lower bound. The following claims are devised to prove this bound.

Claim C.3. *Let $\mathcal{O}$ be a uniform oracle and $R_\mathcal{O}$ be a useful relation. For every relativized interactive argument $\mathsf{ARG} = (\mathcal{P}, \mathcal{V})$ for $R_\mathcal{O}$,*

$$\Pr\left[\begin{array}{l} \mathsf{d} = 0 \\ \wedge 0^n \notin L(R_f) \end{array} \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_{\lambda,n} \\ \mathbb{w}_0 \leftarrow \mathcal{W}_{\lambda,n} \\ f_{(0)} \coloneqq \mathsf{Insert}(f, \mathbb{w}_0) \\ \mathsf{d} \leftarrow \langle \mathcal{P}^{f_{(0)}}(1^\lambda, (0^n, \mathbb{w}_0)), \mathcal{V}^{f_{(0)}}(1^\lambda, 0^n) \rangle \end{array} \right] \leq \alpha(\lambda, n) \ . \tag{6}$$

Proof. We fix $\lambda \in \mathbb{N}$, $n \in \mathbb{N}$. Let $\mathcal{A}(\mathbb{x})$ be as in Lemma B.11. The left-hand side of Eq. 6 is at most

$$\Pr\left[\begin{array}{l} \mathsf{d} = 0 \\ \wedge (0^n, \mathbb{w}_0) \in R_f \end{array} \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_{\lambda,n} \\ \mathbb{w}_0 \leftarrow \mathcal{A}^f(0^n) \\ \mathsf{d} \leftarrow \langle \mathcal{P}^f(1^\lambda, (0^n, \mathbb{w}_0)), \mathcal{V}^f(1^\lambda, 0^n) \rangle \end{array} \right] \tag{7}$$

by Lemma B.11 with $Z = \{0,1\}^{\mathsf{vr}} \times \{0,1\}^{\leq \mathsf{pc}}$, $\phi(f_{(0)}, \mathbb{w}_0, z = (\rho, (b_i)_{i\in[\mathsf{k}]})) = 1 - \mathcal{V}^{f_{(0)}}(1^\lambda, 0^n, (b_i)_{i\in[\mathsf{k}]}; \rho)$,

$$\mathcal{D}_{(f_{(0)},\mathbb{w}_0)} = \left\{ z = (\rho, (b_i)_{i\in[\mathsf{k}]}) \;\middle|\; \mathsf{d} \xleftarrow{\zeta_0,\rho,(b_1,a_1,\ldots,a_\mathsf{k})} \langle \mathcal{P}^{f_{(0)}}(1^\lambda, (0^n, \mathbb{w}_0)), \mathcal{V}^{f_{(0)}}(1^\lambda, 0^n)\rangle \right\}$$

for $(f_{(0)}, \mathbb{w}_0) \in \mathcal{O}_{\lambda,n} \times (\mathcal{W}_{\lambda,n} \cup \{\bot\})$. We conclude the proof by observing that Equation 7 is at most $\alpha(\lambda, n)$ by the completeness of ARG. □

Claim C.4. *For every relativized relation $R_\mathcal{O}$, interactive argument $\mathsf{ARG} = (\mathcal{P}, \mathcal{V})$ for $R_\mathcal{O}$, instance $\mathbb{x} \in \{0,1\}^n$, and $\mathsf{q}_{\widetilde{\mathcal{P}}}$-query $\mathsf{t}_{\widetilde{\mathcal{P}}}$-time adversary $\widetilde{\mathcal{P}}$,*

$$\begin{aligned}&\Pr\left[\begin{array}{l|l} \mathsf{d} = 0 & f \leftarrow \mathcal{O}_{\lambda,n} \\ \mathbb{x} \notin L(R_f) & \mathsf{d} \leftarrow \langle \widetilde{\mathcal{P}}^f(\mathbb{x}), \mathcal{V}^f(1^\lambda, \mathbb{x})\rangle \end{array}\right] \\ &\geq \Pr[\,\mathbb{x} \notin L(R_f) \mid f \leftarrow \mathcal{O}_{\lambda,n}\,] - \beta(\lambda, n, \mathsf{q}_{\widetilde{\mathcal{P}}}, \mathsf{t}_{\widetilde{\mathcal{P}}}) \ .\end{aligned}$$

Proof. We have

$$\begin{aligned}&\Pr\left[\begin{array}{l|l} \mathsf{d} = 0 & f \leftarrow \mathcal{O}_{\lambda,n} \\ \wedge\, \mathbb{x} \notin L(R_f) & \mathsf{d} \leftarrow \langle \widetilde{\mathcal{P}}^f(\mathbb{x}), \mathcal{V}^f(1^\lambda, \mathbb{x})\rangle \end{array}\right] \\ &= \Pr[\,\mathbb{x} \notin L(R_f) \mid f \leftarrow \mathcal{O}_{\lambda,n}\,] - \Pr\left[\begin{array}{l|l} \mathsf{d} = 1 & f \leftarrow \mathcal{O}_{\lambda,n} \\ \wedge\, \mathbb{x} \notin L(R_f) & \mathsf{d} \leftarrow \langle \widetilde{\mathcal{P}}^f(\mathbb{x}), \mathcal{V}^f(1^\lambda, \mathbb{x})\rangle \end{array}\right] \\ &\qquad\qquad \text{(by the law of total probability)} \\ &\geq \Pr[\,\mathbb{x} \notin L(R_f) \mid f \leftarrow \mathcal{O}_{\lambda,n}\,] - \beta(\lambda, n, \mathsf{q}_{\widetilde{\mathcal{P}}}, \mathsf{t}_{\widetilde{\mathcal{P}}}) \ . \qquad \text{(by the soundness of ARG)}\end{aligned}$$

□

The following lemma combines the previous two claims: completeness and soundness together force the verifier to query the witness.

Lemma C.5. *Let $\mathcal{O}$ be a uniform oracle with input size bound ℓ_{in} and $R_\mathcal{O}$ be a useful relation. For every relativized interactive argument $\mathsf{ARG} \coloneqq (\mathcal{P}, \mathcal{V})$ for $R_\mathcal{O}$,*

$$\begin{aligned}&\Pr\left[\begin{array}{l|l} \mathrm{Q}_\mathrm{V}(f_{(0)}, (b'_i)_{i\in[\mathsf{k}]}, \rho, \mathbb{w}_0) = 1 & f \leftarrow \mathcal{O}_{\lambda,n} \\ \wedge\, 0^n \notin L(R_f) & \mathbb{w}_0 \leftarrow \mathcal{W}_{\lambda,n} \\ & f_{(0)} \coloneqq \mathsf{Insert}(f, \mathbb{w}_0) \\ & \mathsf{d}' \xleftarrow{\zeta,\rho,\mathsf{tr}'} \langle \mathcal{P}^{f_{(0)}}(1^\lambda, (0^n, \mathbb{w}_0)), \mathcal{V}^{f_{(0)}}(1^\lambda, 0^n)\rangle \\ & (b'_1, a'_1, \ldots, b'_\mathsf{k}, a'_\mathsf{k}) \coloneqq \mathsf{tr}' \end{array}\right] \\ &\geq \omega(\lambda, n) - \beta(\lambda, n, \mathsf{pq}, O(\mathsf{pt} + \ell_{\mathsf{in}} \cdot \mathsf{pq} + \log|\mathcal{W}_{\lambda,n}|)) - \alpha(\lambda, n) \ .\end{aligned}$$

Proof. We define a malicious prover $\widetilde{\mathcal{P}}$ as follows.

$\widetilde{\mathcal{P}}^f(\mathbb{x})$:

1. Sample $\zeta \leftarrow \{0,1\}^{\mathsf{pr}}$.
2. Sample $\mathbb{w}_0 \leftarrow \mathcal{W}_{\lambda,n}$.
3. For $i \in [\mathsf{k}]$:
   (a) Simulate $b_i \coloneqq \mathcal{P}^{\mathsf{Insert}(f,\mathbb{w}_0)}(1^\lambda, (\mathbb{x}, \mathbb{w}_0), (a_j)_{j\in[i-1]}; \zeta)$ by answering each oracle query q of $\mathcal{P}$ as follows.
      i. Compute $\mathsf{a}' \coloneqq \mathsf{Insert}(f, \mathbb{w}_0)(q)$ by querying f as needed.
      ii. Answer $\mathcal{P}$'s oracle call with a'.
   (b) Output b_i.
   (c) Receive the i-th verifier message a_i.

Since $R_\mathcal{O}$ is a useful relation, $\widetilde{\mathcal{P}}$ can compute $\mathsf{a}' \coloneqq \mathsf{Insert}(f, \mathbb{w}_0)(q)$ for each q in time $O(|q|)$ with 1 query to the oracle. Therefore, $\widetilde{\mathcal{P}}$ has the running time $O(\mathsf{pt} + \ell_{\mathsf{in}} \cdot \mathsf{pq} + \log|\mathcal{W}_{\lambda,n}|)$ and makes at most pq queries to the oracle.

In the rest of this proof, we sample the random variables as in the following random experiment.

$$\left\{\begin{array}{l} f \leftarrow \mathcal{O}_{\lambda,n} \\ \mathbb{w}_0 \leftarrow \mathcal{W}_{\lambda,n} \\ f_{(0)} \coloneqq \mathsf{Insert}(f, \mathbb{w}_0) \\ \mathsf{d} \xleftarrow{\zeta,\rho,\mathsf{tr}} \langle \mathcal{P}^{f_{(0)}}(1^\lambda, (0^n, \mathbb{w}_0)), \mathcal{V}^f(1^\lambda, 0^n)\rangle \\ \mathsf{d}' \xleftarrow{(b'_1,a'_1,\ldots,b'_\mathsf{k},a'_\mathsf{k})} \langle \mathcal{P}^{f_{(0)}}(1^\lambda, (0^n, \mathbb{w}_0); \zeta), \mathcal{V}^{f_{(0)}}(1^\lambda, 0^n; \rho)\rangle \end{array}\right\}$$

Note that $\mathcal{P}^{f_{(0)}}(1^\lambda, (0^n, \mathbb{w}_0); \zeta)$ is equivalent to $\widetilde{\mathcal{P}}^f(0^n)$.

By Definition B.1 and Claim C.4 with $\widetilde{\mathcal{P}}$ defined above, we have

$$\begin{aligned} \omega(\lambda, n) - \beta(\lambda, n, \mathsf{pq}, O(\mathsf{pt} + \ell_{\mathsf{in}} \cdot \mathsf{pq})) &= \Pr[\, 0^n \notin L(R_f)\,] - \beta(\lambda, n, \mathsf{pq}, O(\mathsf{pt} + \ell_{\mathsf{in}} \cdot \mathsf{pq})) \\ &\leq \Pr[\, \mathsf{d} = 0 \wedge 0^n \notin L(R_f)\,] \ . \end{aligned} \tag{8}$$

By invoking Claim C.2 with $\mathcal{D}$ set as the distribution of $(f_{(0)}, f_{(0)}, f_{(0)}, f, (1^\lambda, 0^n, \mathbb{w}_0), (1^\lambda, 0^n), f, \emptyset, Q_{\mathbb{w}_0})$, $A_0 = \mathcal{P}$, $A_1 = \mathcal{V}$, $\phi_1(v) = \mathbb{1}_{\{\mathcal{V} \text{ rejects given the view } v\}}$, and $\phi_2(f) = \mathbb{1}_{\{0^n \notin L(R_f)\}}$, we have

$$\begin{aligned} \Pr[\, \mathsf{d} = 0 \wedge 0^n \notin L(R_f)\,] \leq \Pr[\, \mathsf{d}' = 0 \wedge 0^n \notin L(R_f)\,] \\ + \Pr\left[\, \mathrm{Q_V}(f_{(0)}, (b'_i)_{i\in[\mathsf{k}]}, \rho, \mathbb{w}_0) = 1 \wedge 0^n \notin L(R_f)\,\right] \ . \end{aligned} \tag{9}$$

Finally, by Claim C.3, which follows from the completeness of ARG, we have

$$\Pr[\, \mathsf{d}' = 0 \wedge 0^n \notin L(R_f)\,] \leq \alpha(\lambda, n) \ . \tag{10}$$

Combining Equations 8 to 10 concludes the proof. □

D WITNESS HIDING

Theorem D.1. *Let $t\colon \mathbb{N} \to \mathbb{N}$ be a time-constructible function with $t(n) \geq n$ and $\mathcal{U}$ be the random oracle over $((\{0,1\}^{t(n)+\lceil \log t(n)\rceil+1}, \{0,1\}^\lambda))_{\lambda,n\in\mathbb{N}}$.*

$$\mathsf{NTIME}^{\mathcal{U}}[t] \not\subseteq \mathsf{ARG}^{\mathcal{U}} \left[\begin{array}{lll} \text{completeness error} & \alpha & = 1/10 \\ \text{soundness error} & \beta & = \begin{cases} 1/10 & \mathsf{q}_{\widetilde{\mathcal{P}}} = \mathsf{poly}(\lambda, t(n)), \\ & \mathsf{t}_{\widetilde{\mathcal{P}}} = \mathsf{poly}(\lambda, t(n)) \\ 1 & \textit{otherwise} \end{cases} \\ \text{verifier query bound} & \mathsf{vq} & = \mathsf{poly}(\lambda, t(n)) \\ \text{verifier running time} & \mathsf{vt} & = \mathsf{poly}(\lambda, t(n)) \\ \text{honest prover query bound} & \mathsf{pq} & = \mathsf{poly}(\lambda, t(n)) \\ \text{honest prover time bound} & \mathsf{pt} & = \mathsf{poly}(\lambda, t(n)) \\ \text{witness-hiding error} & \varepsilon_{\mathsf{WH}} & = \begin{cases} 1/10 & \mathsf{aq} = \mathsf{poly}(\lambda, t(n)), \\ & \mathsf{at} = \mathsf{poly}(\lambda, t(n)) \\ 1 & \textit{otherwise} \end{cases} \end{array}\right]$$

Theorem D.1 follows from the following lemma.

Lemma D.2. *Let $\mathcal{O}$ be a uniform oracle with input size bound ℓ_{in} and $R_{\mathcal{O}} \in \mathsf{NTIME}^{\mathcal{O}}[t(n)]$ be a useful relation. If $\mathsf{ARG} = (\mathcal{P}, \mathcal{V})$ is a relativized argument for $R_{\mathcal{O}}$, then the efficiency parameters of ARG satisfy*

$$\alpha + \beta(\mathsf{pq}, O(\mathsf{pt} + \mathsf{pq} \cdot \ell_{\mathsf{in}} + \log|\mathcal{W}_{\lambda,n}|)) + \varepsilon_{\mathsf{WH}}(O(\mathsf{vq} \cdot t(n)), O(\mathsf{vt} + \mathsf{vq} \cdot \ell_{\mathsf{in}} + \mathsf{vq} \cdot t(n))) \geq \omega \ ,$$

where we suppress the common security parameter λ and instance size n for simplicity.

Proof of Theorem D.1. Let $R_{\mathcal{U}}$ be the relation defined as in Definition B.4. By Lemma B.5, $R_{\mathcal{U}}$ has weight $\omega(\lambda, n) = (1 - 1/2^{t(n)})^{2^{t(n)}}$. In particular, $\omega \to 1/e$ as $n \to \infty$ (since $t(n) \geq n$). Therefore, for every sufficiently large n,

$$\alpha + \beta(\mathsf{pq}, O(\mathsf{pt} + \mathsf{pq} \cdot t(n))) + \varepsilon_{\mathsf{WH}}(O(\mathsf{vq} \cdot t(n)), O(\mathsf{vt} + \mathsf{vq} \cdot t(n))) > 0.36 \ ,$$

by Lemma D.2. Since $R_{\mathcal{U}} \in \mathsf{NTIME}^{\mathcal{U}}[t(n)]$ by Lemma B.5, we conclude the proof by observing that the above inequality is not satisfied for a relativized argument for $R_{\mathcal{U}}$ with efficiency parameters as in Theorem D.1. □

D.1 Proof of Lemma D.2

We fix $\lambda, n \in \mathbb{N}$ and suppress them throughout the proof. We use the witness query predicate $\mathrm{Q}_{\mathrm{V}}(\cdot,\cdot,\cdot,\cdot)$ from Definition B.3.

Throughout this proof, we sample the random variables according to the following random experiment.

$$\mathrm{EXP} \coloneqq \left\{ \begin{array}{l} f \leftarrow \mathcal{O}_{\lambda,n} \\ \mathbb{w}_0 \leftarrow \mathcal{W}_{\lambda,n} \\ f_{(0)} \coloneqq \mathsf{Insert}(f, \mathbb{w}_0) \\ \mathsf{d}^{(0)} \xleftarrow{\zeta,\rho,(b_1^{(0)},a_1^{(0)},\ldots,b_{\mathsf{k}}^{(0)},a_{\mathsf{k}}^{(0)})} \langle \mathcal{P}^{f_{(0)}}(1^\lambda,(0^n,\mathbb{w}_0)), \mathcal{V}^{f_{(0)}}(1^\lambda,0^n)\rangle \end{array} \right\} .$$

By Lemma C.5, we are left to prove

$$\Pr\left[\begin{array}{l} \mathrm{Q}_{\mathrm{V}}(f_{(0)},(b_i^{(0)})_{i\in[\mathsf{k}]},\rho,\mathbb{w}_0)=1 \\ \wedge\, 0^n \notin L(R_f) \end{array} \right] \leq \varepsilon_{\mathsf{WH}}(O(\mathsf{vq}\cdot t(n)), O(\mathsf{vt}+\mathsf{vq}\cdot\ell_{\mathsf{in}}+\mathsf{vq}\cdot t(n))) \ . \tag{11}$$

We prove Eq. 11 using the witness-hiding property of ARG. We define witness hiding adversaries $\mathcal{A}_{\mathsf{WH},1}$ and $\mathcal{A}_{\mathsf{WH},2}$ as follows. Let M be the ND-decider for $R_{\mathcal{O}}$ that runs in time $O(t(n))$.

- $\mathcal{A}^f_{\mathsf{WH},1}(\mathbb{x})$:
  1. Compute $\mathsf{Witness}^{R_{\mathcal{O}}}(f,0^n)$ by querying f as needed.
  2. If $\mathsf{Witness}^{R_{\mathcal{O}}}(f,0^n)=\{\mathbb{w}_0\}$ for some $\mathbb{w}_0$, output $\mathbb{w}_0$.
  3. Otherwise, output $\perp$.
- $\mathcal{A}^f_{\mathsf{WH},2}(\mathbb{x},\rho,\mathsf{tr})$:
  1. Parse tr as $(b_1,a_1,\ldots,b_{\mathsf{k}},a_{\mathsf{k}})$
  2. Set $S\coloneqq\emptyset$.
  3. For $i\in\{1,\ldots,\mathsf{k}\}$, simulate $a_i \coloneqq \mathcal{V}^f(1^\lambda,0^n,(b_j)_{j\in[i]};\rho)$ by answering each oracle query q of $\mathcal{V}$ as follows.
     (a) Query f at q; let a be the answer.
     (b) Answer $\mathcal{V}$'s oracle call with a.
     (c) Compute $S' \coloneqq \{\mathbb{w} : q \in Q_{\mathbb{w}}\}$.
     (d) Set $S \coloneqq S' \cup S$.
  4. For $\mathbb{w}\in S$:
     (a) Simulate $\mathsf{out} \coloneqq M^f(\mathbb{x},\mathbb{w})$ by querying the oracle at q and answering M's oracle query with $f(q)$ whenever M queries the oracle at some q.
     (b) If $\mathsf{out}=1$, output $\mathbb{w}$.
  5. Output $\perp$.

We now compute the time and query complexity of $\mathcal{A}_{\mathsf{WH},2}$. The set S' can be computed in time $O(\ell_{\mathsf{in}})$ without querying f. Thus, $\mathcal{A}_{\mathsf{WH},2}$ queries the oracle only to simulate the verifier and the ND-decider. Simulating the verifier requires at most vq queries. Since at most one witness is added to S for each verifier query and the ND-decider runs in time $O(t(n))$, simulating the ND-decider requires at most $\mathsf{vq}\cdot t(n)$ oracle queries. In total, $\mathcal{A}_{\mathsf{WH},2}$ makes at most $\mathsf{vq}+\mathsf{vq}\cdot t(n)$ oracle queries.

Since simulating the verifier takes time $O(\mathsf{vt})$, computing each S' takes time $O(\ell_{\mathsf{in}})$, and simulating M once takes time $O(t(n))$, $\mathcal{A}_{\mathsf{WH},2}$ runs in time $O(\mathsf{vt}+\mathsf{vq}\cdot\ell_{\mathsf{in}}+\mathsf{vq}\cdot t(n))$. Therefore,

$$\Pr\left[(0^n,\mathbb{w}')\in R_f \ \wedge\ (0^n,\mathbb{w}_0)\in R_f \ \middle|\ \begin{array}{l} f\leftarrow\mathcal{O}_{\lambda,n} \\ \mathbb{w}_0\leftarrow\mathcal{A}^f_{\mathsf{WH},1}(0^n) \\ \mathsf{d}\xleftarrow{\zeta,\rho,\mathsf{tr}}\langle\mathcal{P}^f(1^\lambda,(0^n,\mathbb{w}_0)),\mathcal{V}^f(1^\lambda,0^n)\rangle \\ \mathbb{w}'\leftarrow\mathcal{A}^f_{\mathsf{WH},2}(0^n,\rho,\mathsf{tr}) \end{array} \right] \tag{12}$$

is at most $\varepsilon_{\mathsf{WH}}(\lambda,n,O(\mathsf{vq}\cdot t(n)),O(\mathsf{vt}+\mathsf{vq}\cdot\ell_{\mathsf{in}}+\mathsf{vq}\cdot t(n)))$. We are left to show that Equation 12 upper bounds

$$\Pr_{\mathrm{EXP}}\left[\mathrm{Q}_{\mathrm{V}}(f_{(0)},(b_i^{(0)})_{i\in[\mathsf{k}]},\rho,\mathbb{w}_0)=1\ \wedge\ 0^n\notin L(R_f)\right] \ . \tag{13}$$

Equation 13 equals

$$\frac{|\mathcal{W}_{\lambda,n}|-1}{|\mathcal{W}_{\lambda,n}|}\cdot \Pr\left[\begin{array}{l|l} \mathrm{Q}_{\mathrm{V}}(f,(b_i)_{i\in[\mathsf{k}]},\rho,\mathbb{w}_0) & f\leftarrow\mathcal{O}_{\lambda,n} \\ =1\wedge(0^n,\mathbb{w}_0)\in R_f & \mathbb{w}_0\leftarrow\mathcal{A}^f_{\mathsf{WH},1}(0^n) \\ & \mathsf{d}\xleftarrow{\zeta,\rho,\mathsf{tr}}\langle\mathcal{P}^f(1^\lambda,(0^n,\mathbb{w}_0)),\mathcal{V}^f(1^\lambda,0^n)\rangle \\ & \mathbb{w}'\leftarrow\mathcal{A}^f_{\mathsf{WH},2}(0^n,\rho,\mathsf{tr}) \\ & (b_1,a_1,\ldots,b_{\mathsf{k}},a_{\mathsf{k}})\coloneqq\mathsf{tr}\end{array}\right] \tag{14}$$

by Lemma B.11 with $Z=\{0,1\}^{\mathsf{vr}}\times\{0,1\}^{\leq\mathsf{pc}}$, $\phi(f,\mathbb{w}_0,z=(\rho,(b_i^{(0)})_{i\in[\mathsf{k}]}))=\mathrm{Q}_{\mathrm{V}}(f,(b_i^{(0)})_{i\in[\mathsf{k}]},\rho,\mathbb{w}_0)$,

$$\mathcal{D}_{(f,\mathbb{w}_0)}=\left\{z=(\rho,(b_i^{(0)})_{i\in[\mathsf{k}]})\;\middle|\;\mathsf{d}\xleftarrow{\zeta,\rho,(b_1^{(0)},a_1^{(0)},\ldots,b_{\mathsf{k}}^{(0)},a_{\mathsf{k}}^{(0)})}\langle\mathcal{P}^f(1^\lambda,(0^n,\mathbb{w}_0)),\mathcal{V}^f(1^\lambda,0^n)\rangle\right\}$$

for $(f,\mathbb{w}_0)\in\mathcal{O}_{\lambda,n}\times(\mathcal{W}_{\lambda,n}\cup\{\bot\})$. Because $\mathcal{A}_{\mathsf{WH},2}$ outputs a valid witness (one of which is $\mathbb{w}_0$) if the verifier makes a query in $Q_{\mathbb{w}_0}$, Equation 14 is upper bounded by Equation 12. By the equality of Equation 14 and Equation 13, we conclude that Equation 13 is at most Equation 12 and Equation 11 holds.

E DEBATE

We extend Theorem D.1 to debate, the two-prover model of scalable oversight (Irving et al., 2018; Brown-Cohen et al., 2024). We first define relativized debates and witness hiding for them, and then show that a debate whose honest opponent is efficient collapses to a single-prover relativized argument with the same guarantees. The lower bound for debate follows.

Relativized debates. Let $\mathcal{O}$ be an oracle. A ***relativized debate*** in the oracle model $\mathcal{O}$ for a relativized relation $R_{\mathcal{O}}$ is a tuple of interactive oracle algorithms $\mathsf{DEBATE}=(\mathcal{P}_1,\mathcal{P}_2,\mathcal{J})$: the *prover* $\mathcal{P}_1$ receives the witness, the *opponent* $\mathcal{P}_2$ does not, and the *verifier* $\mathcal{J}$ decides. Given security parameter $\lambda\in\mathbb{N}$, instance size $n\in\mathbb{N}$, oracle instance $f\in\mathcal{O}_{\lambda,n}$, instance $\mathbb{x}\in\{0,1\}^n$ and witness $\mathbb{w}\in\{0,1\}^*$, the debate works as follows.

1. Sample the prover randomness ζ_1, the opponent randomness ζ_2, and the verifier randomness ρ.
2. For $i\in[\mathsf{k}]$:
   (a) The prover $\mathcal{P}_1^f(1^\lambda,(\mathbb{x},\mathbb{w}),(b_j,c_j,a_j)_{j<i};\zeta_1)$ computes the message b_i.
   (b) The opponent $\mathcal{P}_2^f(1^\lambda,\mathbb{x},(b_j,c_j,a_j)_{j<i},b_i;\zeta_2)$ computes the message c_i.
   (c) The verifier $\mathcal{J}^f(1^\lambda,\mathbb{x},(b_j,c_j)_{j\leq i};\rho)$ computes the message a_i.
3. The verifier $\mathcal{J}^f(1^\lambda,\mathbb{x},(b_j,c_j)_{j\in[\mathsf{k}]};\rho)$ outputs the decision $\mathsf{d}\coloneqq a_{\mathsf{k}}\in\{0,1\}$.

All messages are public. The opponent speaks after the prover in each round, so it may respond to the prover's message; a debate in which the two provers move simultaneously is the special case in which $\mathcal{P}_2$ ignores b_i when computing c_i. We write $\langle\mathcal{P}_1^f(1^\lambda,(\mathbb{x},\mathbb{w})),\mathcal{P}_2^f(1^\lambda,\mathbb{x}),\mathcal{J}^f(1^\lambda,\mathbb{x})\rangle$ for the decision, and $\mathsf{tr}\coloneqq(b_i,c_i,a_i)_{i\in[\mathsf{k}]}$ for the transcript. The efficiency measures of Section A.3 apply to each party: vq and vt bound the verifier's queries and running time, pq_1 and pt_1 those of the honest prover, and pq_2 and pt_2 those of the honest opponent.

Completeness must hold against every opponent and soundness against every cheating prover, as in (Brown-Cohen et al., 2024).

Definition E.1 (Completeness)**.** DEBATE *has* **completeness error** α *if for every security parameter* $\lambda\in\mathbb{N}$*, instance size* $n\in\mathbb{N}$*, instance* $\mathbb{x}\in\{0,1\}^n$*, adversary* $\mathcal{A}$*, and* q*-query* t*-time opponent* $\widetilde{\mathcal{P}}_2$*,*

$$\Pr\left[\begin{array}{l|l}\mathsf{d}=0 & f\leftarrow\mathcal{O}_{\lambda,n}\\ \wedge(\mathbb{x},\mathbb{w})\in R_f & \mathbb{w}\leftarrow\mathcal{A}^f(\mathbb{x})\\ & \mathsf{d}\leftarrow\langle\mathcal{P}_1^f(1^\lambda,(\mathbb{x},\mathbb{w})),\widetilde{\mathcal{P}}_2^f(\mathbb{x}),\mathcal{J}^f(1^\lambda,\mathbb{x})\rangle\end{array}\right]\leq\alpha(\lambda,n,\mathsf{q},\mathsf{t})\ .$$

Definition E.2 (Soundness)**.** DEBATE *has* **soundness error** β *if for every security parameter* $\lambda\in\mathbb{N}$*, instance size* $n\in\mathbb{N}$*, instance* $\mathbb{x}\in\{0,1\}^n$*, and* q*-query* t*-time cheating prover* $\widetilde{\mathcal{P}}_1$*,*

$$\Pr\left[\begin{array}{l|l}\mathsf{d}=1 & f\leftarrow\mathcal{O}_{\lambda,n}\\ \wedge\mathbb{x}\notin L(R_f) & \mathsf{d}\leftarrow\langle\widetilde{\mathcal{P}}_1^f(\mathbb{x}),\mathcal{P}_2^f(1^\lambda,\mathbb{x}),\mathcal{J}^f(1^\lambda,\mathbb{x})\rangle\end{array}\right]\leq\beta(\lambda,n,\mathsf{q},\mathsf{t})\ .$$

Privacy in a debate must protect the witness from the opponent and the verifier together, since the two may cooperate. We require the weakest such guarantee, in the spirit of Definition A.17: the joint view of the honest opponent and the honest verifier must not allow recovering a witness.

Definition E.3 (Witness hiding)**.** DEBATE *has* **witness-hiding error** $\varepsilon_{\mathsf{WH}}$ *if for every security parameter* $\lambda \in \mathbb{N}$*, instance size* $n \in \mathbb{N}$*, instance* $\mathbb{x} \in \{0,1\}^n$*, adversary* $\mathcal{A}_{\mathsf{WH},1}$*, and* aq*-query* at*-time adversary* $\mathcal{A}_{\mathsf{WH},2}$*, the following is at most* $\varepsilon_{\mathsf{WH}}(\lambda, n, \mathsf{aq}, \mathsf{at})$.

$$\Pr\left[\begin{array}{l|l} (\mathbb{x},\mathbb{w}') \in R_f & f \leftarrow \mathcal{O}_{\lambda,n} \\ \wedge\, (\mathbb{x},\mathbb{w}) \in R_f & \mathbb{w} \leftarrow \mathcal{A}^f_{\mathsf{WH},1}(\mathbb{x}) \\ & \mathsf{d} \xleftarrow{\zeta_1,\zeta_2,\rho,\mathsf{tr}} \langle \mathcal{P}_1^f(1^\lambda,(\mathbb{x},\mathbb{w})), \mathcal{P}_2^f(1^\lambda,\mathbb{x}), \mathcal{J}^f(1^\lambda,\mathbb{x})\rangle \\ & \mathbb{w}' \leftarrow \mathcal{A}^f_{\mathsf{WH},2}(\mathbb{x},\zeta_2,\rho,\mathsf{tr}) \end{array}\right].$$

Remark E.4. The model gives the witness to the prover only. If the opponent also received $\mathbb{w}$, its view would contain $\mathbb{w}$, and no privacy against the opponent would be possible. Requiring witness hiding only against the honest opponent and verifier is the weakest privacy requirement one can ask of a debate; the impossibility below therefore applies to every stronger requirement, such as hiding against a malicious opponent and verifier.

Definition E.5 (Relativized DEBATE in an oracle model)**.** *Let* $\mathcal{O}$ *be an oracle. The complexity class*

$$\mathsf{DEBATE}^{\mathcal{O}}\left[\begin{array}{lccl} \text{completeness error} & \alpha & = & \alpha(\lambda,n,\mathsf{q}_{\widetilde{\mathcal{P}}_2},\mathsf{t}_{\widetilde{\mathcal{P}}_2}) \\ \text{soundness error} & \beta & = & \beta(\lambda,n,\mathsf{q}_{\widetilde{\mathcal{P}}_1},\mathsf{t}_{\widetilde{\mathcal{P}}_1}) \\ \text{verifier query bound} & \mathsf{vq} & = & \mathsf{vq}(\lambda,n) \\ \text{verifier running time} & \mathsf{vt} & = & \mathsf{vt}(\lambda,n) \\ \text{honest prover query bound} & \mathsf{pq}_1 & = & \mathsf{pq}_1(\lambda,n) \\ \text{honest prover time bound} & \mathsf{pt}_1 & = & \mathsf{pt}_1(\lambda,n) \\ \text{honest opponent query bound} & \mathsf{pq}_2 & = & \mathsf{pq}_2(\lambda,n) \\ \text{honest opponent time bound} & \mathsf{pt}_2 & = & \mathsf{pt}_2(\lambda,n) \\ \text{witness-hiding error} & \varepsilon_{\mathsf{WH}} & = & \varepsilon_{\mathsf{WH}}(\lambda,n,\mathsf{aq},\mathsf{at}) \end{array}\right]$$

is the set of relativized relations $R_{\mathcal{O}}$ *that admit a relativized debate* $(\mathcal{P}_1,\mathcal{P}_2,\mathcal{J})$ *with completeness error* α*, soundness error* β*, and witness-hiding error* $\varepsilon_{\mathsf{WH}}$*, in which the verifier makes at most* vq *queries and runs in time* vt*, the honest prover makes at most* pq_1 *queries and runs in time* pt_1*, and the honest opponent makes at most* pq_2 *queries and runs in time* pt_2*.*

Folding the opponent into the verifier. Given a relativized debate, we build a single-prover relativized argument in which the verifier runs the honest opponent internally.

Lemma E.6. *Let* $\mathsf{DEBATE} = (\mathcal{P}_1,\mathcal{P}_2,\mathcal{J})$ *be a relativized debate for* $R_{\mathcal{O}}$ *with completeness error* α*, soundness error* β*, witness-hiding error* $\varepsilon_{\mathsf{WH}}$*, verifier query bound* vq *and running time* vt*, honest prover query bound* pq_1 *and time bound* pt_1*, and honest opponent query bound* pq_2 *and time bound* pt_2*. Let* $\mathcal{V}$ *be the interactive oracle algorithm that, with randomness* (ζ_2,ρ)*, runs* $\mathcal{P}_2$ *and* $\mathcal{J}$ *internally: in round* i*, on receiving* b_i*, it computes* c_i *as* $\mathcal{P}_2$ *would and* a_i *as* $\mathcal{J}$ *would, sends* (c_i,a_i) *to the prover, and at the end outputs the decision of* $\mathcal{J}$*. Then* $(\mathcal{P}_1,\mathcal{V})$ *is a relativized argument for* $R_{\mathcal{O}}$ *with*

- *completeness error* $\alpha(\lambda,n,\mathsf{pq}_2,\mathsf{pt}_2)$*;*
- *soundness error* β*;*
- *witness-hiding error* $\varepsilon_{\mathsf{WH}}$*;*
- *verifier query bound* $\mathsf{vq}+\mathsf{pq}_2$ *and verifier running time* $O(\mathsf{vt}+\mathsf{pt}_2)$*;*
- *honest prover query bound* pq_1 *and honest prover time bound* pt_1*.*

Proof. Fix λ, n, $f \in \mathcal{O}_{\lambda,n}$, $\mathbb{x}$ and $\mathbb{w}$. By construction, the interaction $\langle \mathcal{P}_1^f(1^\lambda,(\mathbb{x},\mathbb{w})), \mathcal{V}^f(1^\lambda,\mathbb{x})\rangle$ with randomness ζ_1 for the prover and (ζ_2,ρ) for the verifier produces the messages $(b_i,c_i,a_i)_{i\in[\mathsf{k}]}$ of the debate $\langle \mathcal{P}_1,\mathcal{P}_2,\mathcal{J}\rangle$ with randomness (ζ_1,ζ_2,ρ), and $\mathcal{V}$ outputs the decision of $\mathcal{J}$ in that debate. The same holds when $\mathcal{P}_1$ is replaced by any cheating prover $\widetilde{\mathcal{P}}_1$, since $\mathcal{V}$ does not depend on the prover. The queries of $\mathcal{V}$ are those of $\mathcal{P}_2$ and $\mathcal{J}$, and its running time is theirs plus the cost of routing messages, which gives the stated bounds. The honest prover is unchanged.

Completeness. The completeness experiment of $(\mathcal{P}_1,\mathcal{V})$ in Definition A.14 is the completeness experiment of DEBATE in Definition E.1 with the opponent $\widetilde{\mathcal{P}}_2 := \mathcal{P}_2$, which makes at most pq_2 queries and runs in time pt_2. Hence the completeness error is at most $\alpha(\lambda,n,\mathsf{pq}_2,\mathsf{pt}_2)$.

Soundness. For a q-query t-time cheating prover $\widetilde{\mathcal{P}}_1$, the soundness experiment of $(\mathcal{P}_1, \mathcal{V})$ in Definition A.15 is the soundness experiment of DEBATE in Definition E.2 against the honest $\mathcal{P}_2$ and $\mathcal{J}$ that $\mathcal{V}$ runs. Hence the soundness error is at most $\beta(\lambda, n, \mathsf{q}, \mathsf{t})$.

Witness hiding. In Definition A.17, the adversary $\mathcal{A}_{\mathsf{WH},2}$ receives the instance, the verifier randomness (ζ_2, ρ), and the transcript $(b_i, (c_i, a_i))_{i\in[\mathsf{k}]}$ of $(\mathcal{P}_1, \mathcal{V})$. This is exactly the input $(\mathbb{x}, \zeta_2, \rho, \mathsf{tr})$ of the adversary in Definition E.3, and the two experiments sample it from the same distribution. Every aq-query at-time adversary against $(\mathcal{P}_1, \mathcal{V})$ is therefore an adversary against DEBATE with the same resources and the same success probability. Hence the witness-hiding error is at most $\varepsilon_{\mathsf{WH}}(\lambda, n, \mathsf{aq}, \mathsf{at})$. □

The lower bound for debate. Combining Lemma E.6 with Theorem D.1 gives the following.

Corollary E.7. *Let $t\colon \mathbb{N} \to \mathbb{N}$ be a time-constructible function with $t(n) \geq n$ and $\mathcal{U}$ be the random oracle over $((\{0,1\}^{t(n)+\lceil \log t(n)\rceil+1}, \{0,1\}^{\lambda}))_{\lambda,n\in\mathbb{N}}$.*

$$\mathsf{NTIME}^{\mathcal{U}}[t] \not\subseteq \mathsf{DEBATE}^{\mathcal{U}} \left[\begin{array}{lcl}
\text{completeness error} & \alpha & = \begin{cases} 1/10 & \mathsf{q}_{\widetilde{\mathcal{P}}_2} = \mathsf{poly}(\lambda, t(n)), \\ & \mathsf{t}_{\widetilde{\mathcal{P}}_2} = \mathsf{poly}(\lambda, t(n)) \\ 1 & \textit{otherwise} \end{cases} \\
\text{soundness error} & \beta & = \begin{cases} 1/10 & \mathsf{q}_{\widetilde{\mathcal{P}}_1} = \mathsf{poly}(\lambda, t(n)), \\ & \mathsf{t}_{\widetilde{\mathcal{P}}_1} = \mathsf{poly}(\lambda, t(n)) \\ 1 & \textit{otherwise} \end{cases} \\
\text{verifier query bound} & \mathsf{vq} & = \mathsf{poly}(\lambda, t(n)) \\
\text{verifier running time} & \mathsf{vt} & = \mathsf{poly}(\lambda, t(n)) \\
\text{honest prover query bound} & \mathsf{pq}_1 & = \mathsf{poly}(\lambda, t(n)) \\
\text{honest prover time bound} & \mathsf{pt}_1 & = \mathsf{poly}(\lambda, t(n)) \\
\text{honest opponent query bound} & \mathsf{pq}_2 & = \mathsf{poly}(\lambda, t(n)) \\
\text{honest opponent time bound} & \mathsf{pt}_2 & = \mathsf{poly}(\lambda, t(n)) \\
\text{witness-hiding error} & \varepsilon_{\mathsf{WH}} & = \begin{cases} 1/10 & \mathsf{aq} = \mathsf{poly}(\lambda, t(n)), \\ & \mathsf{at} = \mathsf{poly}(\lambda, t(n)) \\ 1 & \textit{otherwise} \end{cases}
\end{array}\right]$$

Proof. By Theorem D.1, there exists a relativized relation $R_{\mathcal{U}} \in \mathsf{NTIME}^{\mathcal{U}}[t]$, for which there exists no relativized argument for $R_{\mathcal{U}}$ with parameters as in Theorem D.1. Suppose towards a contradiction that $R_{\mathcal{U}}$ has a relativized debate $(\mathcal{P}_1, \mathcal{P}_2, \mathcal{J})$ with the parameters in the corollary. By Lemma E.6, $(\mathcal{P}_1, \mathcal{V})$ is a relativized argument for $R_{\mathcal{U}}$ with completeness error $\alpha(\lambda, n, \mathsf{pq}_2, \mathsf{pt}_2) = 1/10$, because pq_2 and pt_2 are polynomial in λ and $t(n)$; with soundness error $1/10$ against $\mathsf{poly}(\lambda, t(n))$-query $\mathsf{poly}(\lambda, t(n))$-time cheating provers; with witness-hiding error $1/10$ against $\mathsf{poly}(\lambda, t(n))$-query $\mathsf{poly}(\lambda, t(n))$-time adversaries; with verifier query bound $\mathsf{vq} + \mathsf{pq}_2 = \mathsf{poly}(\lambda, t(n))$ and verifier running time $O(\mathsf{vt} + \mathsf{pt}_2) = \mathsf{poly}(\lambda, t(n))$; and with honest prover query and time bounds $\mathsf{poly}(\lambda, t(n))$. Since $(\mathcal{P}_1, \mathcal{V})$ contradicts that there exists no relativized argument for $R_{\mathcal{U}}$ with parameters as in Theorem D.1, we conclude $R_{\mathcal{U}}$ has no relativized debate $(\mathcal{P}_1, \mathcal{P}_2, \mathcal{J})$ with the parameters in the corollary. □

Remark E.8. The proof of Lemma E.6 uses completeness only against the honest opponent, so Corollary E.7 holds even for debates whose completeness is required only against the honest opponent. The folding requires the honest opponent to be efficient, as in doubly-efficient debate (Brown-Cohen et al., 2024); a debate whose honest opponent runs in more than polynomial time is not covered. Finally, Corollary E.7 concerns witness hiding. Zero knowledge against the opponent and the verifier implies witness hiding up to the probability of finding a witness from the instance alone, which is negligible for $R_{\mathcal{U}}$; we do not formalize this implication here.

F Zero-knowledge verification of oracle-aided computations

Theorem D.1 rules out privacy for relativized arguments in the random oracle model. In this section, we study arguments for oracle circuit satisfiability, which we are able to construct by requiring oracle to give signed answers in Section F.1.

Oracle-aided computation.

We use the convention that a boolean circuit is a circuit $\mathcal{C}$ with gates $\wedge, \vee, \neg$ such that $\mathcal{C}$ accepts a fixed-length binary string as input and outputs a binary string. We define $\mathcal{C}$'s output on binary strings with wrong length to be $\perp$.

Definition F.1. *Let $f\colon \{0,1\}^{\ell_{\mathsf{in}}} \to \{0,1\}$ be a function. An (adaptive)* **oracle (boolean) circuit** *$\mathcal{C}^f$ with input size n is a tuple of boolean circuits $(\mathcal{C}_0, \ldots, \mathcal{C}_{\mathsf{q}})$ for some $\mathsf{q} \in \mathbb{N}$, where for $i \in [\mathsf{q}+1]$, $\mathcal{C}_{i-1}\colon \{0,1\}^{n+i-1} \to \{0,1\}^{\ell_{\mathsf{in}}}$, except that $\mathcal{C}_{\mathsf{q}}$ outputs a single bit. On input $\mathbb{x} \in \{0,1\}^n$, the output of $\mathcal{C}^f(\mathbb{x})$ is defined to be $\mathcal{C}_{\mathsf{q}}(\mathbb{x}\|\mathsf{a}_1\|\ldots\|\mathsf{a}_{\mathsf{q}})$, where for $i \in [\mathsf{q}]$, $q_i \coloneqq \mathcal{C}_{i-1}(\mathbb{x}\|\mathsf{a}_1\|\ldots\|\mathsf{a}_{i-1})$ and $\mathsf{a}_i \coloneqq f(q_i)$. We call $(q_1, \mathsf{a}_1, \ldots, q_{\mathsf{q}}, \mathsf{a}_{\mathsf{q}})$ the query-answer trace of $\mathcal{C}^f(\mathbb{x})$. We use an explicit description $\langle\mathcal{C}\rangle = (\langle\mathcal{C}_0\rangle, \ldots, \langle\mathcal{C}_{\mathsf{q}}\rangle)$. There exists an algorithm that evaluates any such description in time polynomial in its length and the input length.*

We are interested in constructing a zero-knowledge argument for the following relativized circuit satisfiability relation.

Definition F.2. *For every oracle $\mathcal{O}$, the relativized relation $R^{\mathsf{OCSAT}}_{\mathcal{O}} = \{R^{\mathsf{OCSAT}}_f\}_{f\in\mathcal{O}}$ is defined as follows for $f \in \mathcal{O}$:*

$$R^{\mathsf{OCSAT}}_f = \{(\mathbb{x} = (\langle\mathcal{C}\rangle, \mathbb{x}'), \mathbb{w})\colon \mathcal{C}^f(\mathbb{x}'\|\mathbb{w}) = 1\} \ .$$

F.1 Zero-knowledge verification with a signed oracle

We authenticate oracle answers using the signature-card approach of Chiesa & Tromer (2010). We consider an oracle model in which a trusted party authenticates every oracle answer. Informally, a trusted party holds a signing key and, on a query q, returns the oracle answer $\mathsf{a} = f(q)$ together with a signature on the query–answer pair (q, a).

Definition F.3. *Let $\mathcal{O}$ be an oracle and let $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ be a digital signature scheme as in Definition A.21. The* **signed oracle** *associated with $\mathcal{O}$ and SIG, denoted $\widehat{\mathcal{O}}$, is the collection $\{\widehat{\mathcal{O}}_{\lambda,n}\}_{\lambda,n\in\mathbb{N}}$, where $\widehat{\mathcal{O}}_{\lambda,n}$ is the following distribution.*

$$\left\{ \widehat{f} \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_{\lambda,n} \\ (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^\lambda) \\ \text{For } q \in X_{\lambda,n}\colon \sigma_q \leftarrow \mathsf{Sign}(\mathsf{sk}, (q, f(q))) \\ \text{Let } \widehat{f} \text{ be the function defined by } \widehat{f}(q) \coloneqq \begin{cases} (f(q), \sigma_q) & \text{if } q \in X_{\lambda,n} \\ \mathsf{vk} & \text{if } q = \text{“key”} \end{cases} \end{array} \right\} .$$

Above, $X_{\lambda,n}$ is the domain of f and we assume "key" $\notin X_{\lambda,n}$. We call f the underlying oracle of $\widehat{f}$.

The signed oracle model can be implemented in practice with a trusted party who has access to the oracle and lazily samples the signed oracle. The signed oracle model is deliberately minimal: a trusted party performs no computation beyond signing the query–answer pair. It does, however, *see the queries*, which in our applications contain private data. In particular, the trusted party is trusted to issue the signatures correctly, to not issue a signature for an invalid query–answer pair, and keep the secret signing key and oracle queries private.

For a relativized relation $R_{\mathcal{O}}$, we define its ***signed-oracle lift*** $R_{\widehat{\mathcal{O}}}$ by preserving the relation of the underlying oracle: for every signed-oracle instance $\widehat{f}$ with underlying oracle f,

$$R_{\widehat{f}} \coloneqq R_f \ .$$

Thus, the signatures authenticate oracle answers, but do not change which instance-witness pairs belong to the relation. We emphasize that the signed oracle $\widehat{\mathcal{O}}$ is an oracle according to Definition A.2 and the relation $R_{\widehat{\mathcal{O}}}$ is a relativized relation according to Definition A.4.

We prove that there exist a zero-knowledge argument for $R^{\mathsf{OCSAT}}_{\widehat{\mathcal{O}}}$.

Theorem F.4. *Let $\mathcal{O}$ be an oracle with one-bit answers. We assume that there exists a CRH family that is secure in the plain model and relative to f for all $f \in \mathcal{O}$. Let $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ be a perfectly correct digital signature scheme with EUF-CMA security relative to $\mathcal{O}$ and $\widehat{\mathcal{O}}$ be the signed oracle associated with $\mathcal{O}$ and SIG.*

There exists a relativized argument ARG for the signed-oracle lift $R^{\mathsf{OCSAT}}_{\widehat{\mathcal{O}}}$ of $R^{\mathsf{OCSAT}}_{\mathcal{O}}$ with perfect completeness, soundness $1/2 + \mathrm{negl}(\lambda, \mathrm{n})$, communication $\mathsf{poly}(\lambda, n)$, and verifier time $\mathsf{poly}(\lambda, n)$. ARG

has malicious-verifier statistical zero-knowledge error at most $2^{-\lambda}$*, with simulator time* $\mathsf{poly}(\lambda, n, t_V)$ *and query bound* $\mathsf{poly}(\lambda, n, q_V)$ *against* t_V*-time* q_V*-query malicious verifiers.*

Theorem F.4 follows from the following compiler, Corollary A.23, and Theorem F.6. We omit the proof that CRH security for $\mathcal{O}$ extends to $\widehat{\mathcal{O}}$.

Theorem F.5. *Let* $\mathcal{O}$ *be an oracle with one-bit answers and* $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *be a perfectly correct digital signature scheme with EUF-CMA security relative to* $\mathcal{O}$*. Let* $\widehat{\mathcal{O}}$ *be the signed oracle associated with* $\mathcal{O}$ *and* SIG*. Let* $M(\lambda, n)$ *be such that* $M(\lambda, n) \geq \lambda + n + |\mathsf{vk}|$ *for every* vk *generated by* $\mathsf{Gen}(1^\lambda)$*.*

Suppose for every relation in NP*, there exists a perfectly complete relativized argument in the model* $\widehat{\mathcal{O}}$ *that has knowledge error* κ *with extractor time* $\mathsf{et}(\lambda, n, \mathsf{at}, \delta) = \mathsf{poly}(\lambda, n, \mathsf{at}, 1/\delta)$*, malicious-verifier zero-knowledge with error* z_0 *and simulator time* st_0*, query* sq_0*, prover communication complexity* pc*, and verifier time* vt*.*

There exists a relativized argument for the signed-oracle lift $R^{\mathsf{OCSAT}}_{\widehat{\mathcal{O}}}$ *of* $R^{\mathsf{OCSAT}}_{\mathcal{O}}$ *with perfect completeness, soundness error* β*, malicious-verifier zero-knowledge error* z *with simulator time* st *and query* sq*, prover communication complexity* pc *and verifier time* $O(\mathsf{vt} + M(\lambda, n))$*, such that* $\beta(\lambda, n, \mathsf{q}_{\widetilde{\mathcal{P}}}, \mathsf{t}_{\widetilde{\mathcal{P}}}) = \kappa(\lambda, M(\lambda, n), \mathsf{q}_{\widetilde{\mathcal{P}}}, \mathsf{t}_{\widetilde{\mathcal{P}}}) + \mathrm{negl}(\lambda, \mathrm{n})$ *if* $\mathsf{t}_{\widetilde{\mathcal{P}}} = \mathsf{poly}(\lambda, n)$*,* $z(\lambda, n, \mathsf{aq}, \mathsf{at}) = z_0(\lambda, M(\lambda, n), \mathsf{aq}, \mathsf{at})$*,* $\mathsf{st}(\lambda, n, \mathsf{aq}, \mathsf{at}) = O(\mathsf{st}_0(\lambda, M(\lambda, n), \mathsf{aq}, \mathsf{at}))$*,* $\mathsf{sq}(\lambda, n, \mathsf{aq}, \mathsf{at}) = \mathsf{sq}_0(\lambda, M(\lambda, n), \mathsf{aq}, \mathsf{at}) + 1$*.*

Proof. All inner resource bounds below are evaluated at length $M(\lambda, n)$. The inner argument uses the same signed-oracle function sampled at the outer parameters (λ, n).

We use fixed polynomial bounds on signature lengths for a given security parameter and message length. Malformed encodings and components exceeding these bounds are rejected. For an oracle-circuit instance $\mathbb{x} = (\langle \mathcal{C} \rangle, \mathbb{x}')$, write $\mathcal{C} = (\mathcal{C}_0, \ldots, \mathcal{C}_{\mathsf{q}})$ and define the ordinary relation

$$R_{\mathsf{sig}} := \left\{ \left((1^\lambda, \mathbb{x}, \mathsf{vk}), (\mathbb{w}, (q_i, a_i, \sigma_i)_{i \in [\mathsf{q}]})\right) \;\middle|\; \begin{array}{l} \mathcal{C}_{\mathsf{q}}(\mathbb{x}' \| \mathbb{w} \| a_1 \| \ldots \| a_{\mathsf{q}}) = 1, \\ \forall i \in [\mathsf{q}] : \; q_i = \mathcal{C}_{i-1}(\mathbb{x}' \| \mathbb{w} \| a_1 \| \ldots \| a_{i-1}), \\ \forall i \in [\mathsf{q}] : \; \mathsf{Verify}(\mathsf{vk}, \langle q_i, a_i \rangle, \sigma_i) = 1 \end{array} \right\}. \tag{15}$$

The explicit circuit bounds the witness length, number of queries, and query lengths polynomially in n. Together with the signature-length bounds and the unary security parameter, this gives an NP-verifier and polynomial witness length for R_{sig}, so $R_{\mathsf{sig}} \in \mathsf{NP}$. The NP verifier has no oracle gates: membership asserts that the trace is signed, not that its answers agree with f.

On common input $(1^\lambda, \mathbb{x})$ and prover witness $\mathbb{w}$, we define the interaction between the prover and the verifier as follows.

1. Both parties query $\widehat{f}(\text{“}key\text{''})$ and set y to the padded encoding of $(1^\lambda, \mathbb{x}, \mathsf{vk})$.
2. For $i = 1, \ldots, \mathsf{q}$, the prover computes $q_i = \mathcal{C}_{i-1}(\mathbb{x}' \| \mathbb{w} \| a_1 \| \ldots \| a_{i-1})$ and obtains $(a_i, \sigma_i) = \widehat{f}(q_i)$. It forms $w_{\mathsf{sig}} = (\mathbb{w}, (q_i, a_i, \sigma_i)_i)$.
3. The parties run the argument assumed in the theorem for R_{sig} on y, with prover witness w_{sig}. The verifier outputs the inner decision.

For a valid outer witness, the trace follows the true computation and every signature verifies, so $(y, w_{\mathsf{sig}}) \in R_{\mathsf{sig}}$. Perfect completeness follows. No additional messages are sent, and the verifier's additional work is the key query and the $O(M(\lambda, n))$-time construction of y, giving the communication and verifier-time bounds.

Soundness. Fix a malicious outer prover and let $\varepsilon_{\widehat{f}}$ be its acceptance probability for a fixed signed oracle. Write $\kappa_{\widehat{f}}$ for the inner knowledge error at the padded length and the wrapped prover's resources. When knowledge error is specified pointwise as in Definition A.20, the oracle-independent bound in the theorem is an upper bound on $\mathbb{E}_{\widehat{f}}[\kappa_{\widehat{f}}]$.

For any promised gap $\delta \in (0, 1]$, an EUF-CMA adversary B_δ, given the verification key and access to f and the signing oracle, proceeds as follows.

1. Simulate $\widehat{f}$ lazily: answer key queries with vk; on the first domain query q, compute $a = f(q)$, request a signature on $\langle q, a \rangle$, and cache the pair permanently. Handle all other queries according to Definition A.10, and retain the cache through every rewind.

2. Run the inner extractor on y and the wrapped malicious prover with promised gap δ, using the same simulated signed oracle for all executions and stopping after its polynomial time bound. Check its output using the NP verifier for R_{sig} and return $\perp$ on timeout or an invalid output.
3. Parse a valid output as $(\mathbb{w}, (q_i, a_i, \sigma_i)_i)$. Query $f(q_i)$ until finding $a_i \neq f(q_i)$, and output $(\langle q_i, a_i\rangle, \sigma_i)$. Return $\perp$ if none exists.

The lazy simulation has exactly the signed-oracle distribution. If $\mathbb{x}$ is false and $\varepsilon_{\widehat{f}} > \kappa_{\widehat{f}} + \delta$, knowledge soundness gives a valid extracted trace with probability at least $2/3$. Such a trace contains an incorrect answer: otherwise induction over the circuit queries gives a genuinely accepting computation. The signature on an incorrect answer is a forgery, because only pairs $\langle q, f(q)\rangle$ were submitted for signing. Splitting according to whether $\varepsilon_{\widehat{f}} > \kappa_{\widehat{f}} + \delta$ therefore gives

$$\Pr[\text{accept} \ \wedge \ \mathbb{x} \notin L(R_f^{\mathsf{OCSAT}})] \leq \mathbb{E}_{\widehat{f}}[\kappa_{\widehat{f}}] + \delta + \frac{3}{2}\Pr[B_\delta \text{ forges}].$$

Each new domain query costs one underlying query and one signing query, and checking the extracted trace costs at most q further underlying queries. Thus B_δ is polynomial-time for every inverse-polynomial δ and polynomially bounded instance sizes and adversaries. Since the displayed bound holds for every such δ, EUF-CMA security makes the excess over the inner knowledge-error bound negligible, as claimed.

Zero knowledge. Fix a signed-oracle function and a valid outer instance–witness pair. Define the simulator as follows.

1. Query $\widehat{f}(\text{“}key\text{''})$ and construct the same padded auxiliary instance y.
2. Run the assumed inner simulator on y with a wrapped verifier that runs the original malicious verifier on $\mathbb{x}$, forwarding all messages and oracle queries unchanged.
3. Replace the auxiliary public-input field of the returned view by $\mathbb{x}$ and return it; propagate simulation failure unchanged.

The fixed witness and signed trace satisfy $(y, w_{\mathsf{sig}}) \in R_{\mathsf{sig}}$. The same deterministic view conversion maps the real inner interaction to the real outer interaction, so statistical distance does not increase. The initial key query is the only additional oracle query. Black-box rewinding and completed-view access are inherited from the inner simulator. This proves the privacy and resource claims. □

F.2 A ZERO-KNOWLEDGE SUCCINCT ARGUMENT RELATIVE TO AN ORACLE

We apply a vector-commitment formulation of Kilian's compiler (Kilian, 1992) and (Barak & Goldreich, 2008) to the HVZK PCP of knowledge from Theorem A.26. The relation is non-relativized. We assume only malicious parties, the extractor, and the simulator access the oracle. The simulator below upgrades the PCP's honest-verifier guarantee to malicious-verifier zero knowledge by rewinding.

Theorem F.6. *Let R be an* NP *relation with a deterministic verifier that runs in time $T(n) \geq \max\{n, 2\}$. Let $\mathcal{O}$ be an oracle. We assume that there exists a CRH family that is secure in the plain model and relative to f for all $f \in \mathcal{O}$.*

There exists a relativized argument $\mathsf{KZK} = (\mathcal{P}_{\mathsf{K}}, \mathcal{V}_{\mathsf{K}})$ with perfect completeness, knowledge soundness error κ, malicious-verifier zero-knowledge error z, communication $\mathsf{poly}(\lambda, \log T(n))$, prover time $\mathsf{poly}(\lambda, T(n))$, and verifier time $\mathsf{poly}(\lambda, n, \log T(n))$ such that the following holds. Neither honest party queries the oracle. The error $\kappa(\lambda, n, \mathsf{aq}, \mathsf{at})$ is at most $1/2 + \mathrm{negl}(\lambda)$ if all of $n, \mathsf{aq}, \mathsf{at}$ is at most $\mathsf{poly}(\lambda)$. For every gap $\delta \in (0, 1]$, the knowledge extractor time is $\mathsf{poly}(\lambda, T(n), \mathsf{at}, 1/\delta)$. The error $z(\lambda, n, \mathsf{aq}, \mathsf{at})$ is at most $2^{-\lambda}$ and the simulator runs in time $\mathsf{poly}(\lambda, T(n), \mathsf{at})$ and has query complexity at most $\mathsf{poly}(\lambda, T(n), \mathsf{aq})$.

Proof of Theorem F.6. Let PCP be as in Theorem A.26 with proof length N, randomness complexity a, and the decoding threshold $s \leq 1/2$. We set $M := 2^a$, $L := 2M(\lambda + 3)$, and $\tau := \lambda + 3 + \lceil \log L \rceil$. Let VC be as in Theorem A.29 with message space $\{0, 1\}$ and message-size bound 1.

Protocol F.7. On common input $(1^\lambda, \mathbb{x})$ and prover witness $\mathbb{w}$, the prover and verifier interacts as follows.

1. The verifier samples $\mathsf{pp}_{\mathsf{vc}} \leftarrow \mathsf{VC.Gen}(1^\lambda, 1^\tau, N, 1)$ and sends $\mathsf{pp}_{\mathsf{vc}}$.

2. The prover checks $\mathsf{Adm}_{\mathsf{VC}}(1^\lambda, 1^\tau, N, 1, \mathsf{pp}_{\mathsf{vc}}) = 1$ and otherwise aborts. It samples $\pi_{\mathsf{pcp}} \leftarrow \mathcal{P}_{\mathsf{pcp}}(\mathbb{x}, \mathbb{w})$, computes $(c, \mathsf{st}) \leftarrow \mathsf{VC.Commit}(\mathsf{pp}_{\mathsf{vc}}, \pi_{\mathsf{pcp}})$, and sends c.
3. The verifier samples $r \leftarrow \{0,1\}^a$ uniformly and sends r.
4. For every distinct index i in $I_{\mathbb{x},r}$, ordered increasingly, the prover samples $\pi_{\mathsf{vc}}^{(i)} \leftarrow \mathsf{VC.Open}(\mathsf{pp}_{\mathsf{vc}}, \mathsf{st}, i)$ and sends $(\pi_{\mathsf{pcp}}[i], \pi_{\mathsf{vc}}^{(i)})$. The verifier accepts if $\mathsf{VC.Check}(\mathsf{pp}_{\mathsf{vc}}, c, i, \pi_{\mathsf{pcp}}[i], \pi_{\mathsf{vc}}^{(i)}) = 1$ for every opened position and $D_{\mathbb{x},r}$ accepts the resulting ordered tuple of bits, reusing the bit at any repeated index.

Both parties enforce the correct message formats and length bounds; malformed or missing messages cause rejection or abort. The prover aborts on any challenge outside $\{0,1\}^a$. A fixed empty initial prover message embeds this verifier-first protocol in the convention of Section A.3.

Perfect completeness follows from PCP completeness and perfect VC completeness. By Theorem A.26, $N, M = \mathsf{poly}(T(n))$ and the number of opened positions is constant. The succinct VC parameter, commitment, and opening lengths, together with the bounds on $\mathsf{VC.Gen}$, $\mathsf{Adm}_{\mathsf{VC}}$, and $\mathsf{VC.Check}$ from Definition A.28, give the communication and verifier-time bounds. The PCP and VC algorithms give the prover-time bound.

Knowledge extraction. Given the promised gap δ, the extractor $\mathcal{E}_{\mathsf{K}}^{f,\widetilde{\mathcal{P}_{\mathsf{K}}}}(1^\lambda, \mathbb{x}, \delta)$ performs the following algorithm.

1. Repeat the following trial independently $J = \lceil 3/\delta \rceil$ times, stopping on a verified witness.
   (a) Generate fresh honest $\mathsf{pp}_{\mathsf{vc}} \leftarrow \mathsf{VC.Gen}(1^\lambda, 1^\tau, N, 1)$, start a fresh prover execution, send the parameters, and receive its commitment c. Discard this trial if the commitment is malformed or absent. Otherwise checkpoint the prover before the challenge and initialize an empty table of opened positions.
   (b) For each $r \in \{0,1\}^a$, restore that checkpoint, send r, receive the response, and run the honest verification procedure. Ignore rejecting responses. Insert the records from accepting responses into the table. If two records assign distinct bits b, b' to the same position i, discard this trial; their openings $\pi_{\mathsf{vc}}, \pi'_{\mathsf{vc}}$ give a VC position-binding violation $(c, i, b, b', \pi_{\mathsf{vc}}, \pi'_{\mathsf{vc}})$ under $\mathsf{pp}_{\mathsf{vc}}$.
   (c) If the trial was not discarded, form $\pi \in \{0,1\}^N$ from the table, filling unrecorded positions with zero. Run $\mathcal{D}_{\mathsf{pcp}}(\mathbb{x}, \pi)$ with fresh coins. Return its output if it passes the checker for R.
2. Return $\perp$.

The above extractor runs in time $\mathsf{poly}(\lambda, T(n), \mathsf{at})$ in one repetition, resulting with total time complexity $\mathsf{poly}(\lambda, T(n), \mathsf{at}) \cdot 1/\delta$. Let $\kappa_{\mathsf{K}} \coloneqq \kappa_{\mathsf{dec}} + \beta_{\mathsf{vc}}$. We prove that KZK has knowledge error κ_{K}. Since $\kappa_{\mathsf{dec}} \le 1/2$ and β_{vc} is $\mathrm{negl}(\lambda)$ if all of $n, \mathsf{aq}, \mathsf{at}$ is at most $\mathsf{poly}(\lambda)$, the knowledge soundness in the theorem follows.

For the analysis, let Z include the fresh parameters and the complete prover tape in one trial; the extractor need not inspect the tape. Define $A_Z(r)$ to be acceptance of the continuation with challenge r, setting all these bits to zero if the prefix is invalid, and let $u(Z) = M^{-1} \sum_r A_Z(r)$. If ε is the malicious prover's acceptance probability, then

$$\mathbb{E}_Z[u(Z)] = \varepsilon, \qquad \Pr[u(Z) > s] \ge \varepsilon - s. \tag{16}$$

The latter follows from $u \le s + \mathbf{1}_{\{u>s\}}$. Let B be the event that the table contains conflicting bits. Each occurrence of B yields a VC position-binding violation. The reduction B_{vc} receives fresh $\mathsf{pp}_{\mathsf{vc}}$ from its binding game and runs the exhaustive trial using these parameters, forwarding all queries to f. On B it outputs the two conflicting valid openings. Consequently,

$$\Pr[B] \le \beta_{\mathsf{vc}} \qquad \Pr[u(Z) > s \ \wedge \ \neg B] \ge \varepsilon - s - \beta_{\mathsf{vc}} \ ,$$

where β_{vc} is $\beta_{\mathsf{vc}}^f(\lambda, \tau, N, 1, \mathsf{poly}(\lambda, T(n), \mathsf{at}), \mathsf{poly}(\lambda, T(n), \mathsf{at}))$. On $\neg B$, every accepting branch supplies exactly the bits used by its PCP predicate in the reconstructed word. Hence $\mathsf{acc}_{\mathsf{pcp}}(\mathbb{x}, \pi) \ge u(Z)$. When $\varepsilon \ge \kappa_{\mathsf{K}} + \delta$, one trial returns a witness with probability at least $2\delta/3$. Independent trials fail with probability at most $(1 - 2\delta/3)^J \le e^{-2} < 1/3$. This proves knowledge soundness and the stated worst-case bounds.

Zero-knowledge simulation. Fix f, $(\mathbb{x}, \mathbb{w}) \in R$, and a malicious verifier. The simulator $\mathcal{S}_{\mathsf{K}}^{f,\widetilde{\mathcal{V}}}(1^\lambda, \mathbb{x})$ is as follows.

1. Start one fresh verifier execution and obtain its parameter message. If it is absent or inadmissible, follow the honest prover's abort behavior and return the completed verifier view. Otherwise checkpoint the verifier immediately after this message.
2. Perform at most L trials. In each trial restore the checkpoint and sample $b \leftarrow \{0,1\}$.
   (a) If $b = 0$, sample $(r, v) \leftarrow \mathcal{S}_{\mathsf{pcp}}(\mathbb{x})$, set the guess $g = r$, and let $\widetilde{\pi}$ agree with v at the indices $I_{\mathbb{x},r}$ and be zero elsewhere. If $b = 1$, sample $u \leftarrow \{0,1\}^a$; skip this trial unless $u = 0^a$, and otherwise set $g = \perp$ and $\widetilde{\pi} = 0^N$.
   (b) Compute $(c, \mathsf{st}) \leftarrow \mathsf{VC.Commit}(\mathsf{pp}_{\mathsf{vc}}, \widetilde{\pi})$ and send c. Normalize the verifier's response to $\widehat{r} \in \{0,1\}^a \cup \{\perp\}$, using $\widehat{r} = \perp$ for a malformed challenge, abort, or termination without a challenge.
   (c) If $\widehat{r} \neq g$, discard this continuation without requesting its view. If $\widehat{r} = g \in \{0,1\}^a$, send $(\widetilde{\pi}[i], \mathsf{VC.Open}(\mathsf{pp}_{\mathsf{vc}}, \mathsf{st}, i))$ for each distinct index in $I_{\mathbb{x},\widehat{r}}$, in increasing order, let the verifier finish, and return its completed view. If $\widehat{r} = g = \perp$, follow the honest abort behavior if needed and return the completed view.
3. Return $\perp$ if no trial returned a view.

The simulator runs in time $\mathsf{poly}(\lambda, T(n), \mathsf{at})$ and has query complexity at most $\mathsf{poly}(\lambda, T(n), \mathsf{aq})$.

The padding branch makes every legal guess and the abort guess have probability exactly $1/(2M)$, using only a fixed number of fair coins. Guessing uniformly from a non-power-of-two set is unnecessary. Perfect HVZK ensures that the simulated answer tuple is consistent at repeated indices.

For comparison only, define an ideal simulator using $\mathbb{w}$: retain the same padding and guessing distribution, but in every non-skipped trial commit to a fresh honest PCP proof, independent of the guess, and answer matched challenges honestly. Fix the verifier's entire tape and admissible prefix. In this ideal experiment the guess is independent of the honest continuation, so every possible normalized challenge is matched with probability exactly $1/(2M)$. Thus the success probability is $1/(2M)$ for every such prefix and tape, and conditional on a match the completed view has exactly the real distribution. The abort guess is needed for this statement. The truncated ideal simulator differs from the real interaction by at most $(1 - 1/(2M))^L \leq e^{-L/(2M)}$.

We compare one actual trial with one ideal trial at any fixed saved verifier state. For a legal guess, Eq. 2 identifies the joint distribution of the guessed coins and the opened bits with that of an honest proof and independent uniform coins. Condition on those coins and bits. Each honest proof in this conditional distribution agrees with $\widetilde{\pi}$ on the guessed positions, so $\eta := \eta_{\mathsf{vc}}(\lambda, \tau, N, 1)$ bounds the distance between the joint distributions of the VC commitment and those openings by η. The verifier's subsequent challenge, the match test, and the matched continuation are common postprocessing with the same fixed oracle. For an abort guess, use VC's hiding with no opened positions. Therefore a trial changes statistical distance by at most η. A hybrid over at most L trials, including their discard decisions, proves

$$\mathrm{SD}(\text{real view}, \mathcal{S}_{\mathsf{K}}^{f,\widetilde{\mathcal{V}}}(1^\lambda, \mathbb{x})) \leq L\eta + e^{-L/(2M)} \leq 2^{-\lambda-2}.$$

□