Relativized Succinct Arguments in the ROM Do Not Exist

Annalisa Barbara
annalisa.barbara@studbocconi.it
Bocconi University

Alessandro Chiesa
alessandro.chiesa@epfl.ch
EPFL

Ziyi Guan
ziyi.guan@epfl.ch
EPFL

May 12, 2024

Abstract

A relativized succinct argument in the random oracle model (ROM) is a succinct argument in the ROM that can prove/verify the correctness of computations that involve queries to the random oracle. We prove that relativized succinct arguments in the ROM do not exist. The impossibility holds even if the succinct argument is interactive, and even if soundness is computational (rather than statistical).

This impossibility puts on a formal footing the commonly-held belief that succinct arguments require non-relativizing techniques. Moreover, our results stand in sharp contrast with other oracle models, for which a recent line of work has constructed relativized succinct non-interactive arguments (SNARGs). Indeed, relativized SNARGs are a powerful primitive that, e.g., can be used to obtain constructions of IVC (incrementally-verifiable computation) and PCD (proof-carrying data) based on falsifiable cryptographic assumptions. Our results rule out this approach for IVC and PCD in the ROM.

Keywords: succinct arguments; relativization; random oracle model

Contents

1 Introduction

Succinct non-interactive arguments (SNARGs) are short cryptographic proofs that admit fast verification. SNARGs of knowledge (SNARKs) can be used to construct incrementally-verifiable computation (IVC) [Val08] and proof-carrying data (PCD) [CT10], which are powerful cryptographic primitives that enable efficient verification of distributed computations. Specifically, IVC and PCD can be obtained by proving the security of *recursive proof composition* applied to the underlying SNARK.

Unfortunately, the security analysis of recursive proof composition is asymptotically and concretely expensive. This is due to the fact that the knowledge soundness property of a SNARK is typically established assuming certain knowledge assumptions in the standard model (with no oracles), which introduce expensive blowups. The result is that the security analysis works only for a constant number of recursions [BCCT13; BCTV14; COS20] because each invocation of the underlying SNARK extractor incurs a polynomial blow-up in size/time relative to the prior invocation (leading to an exponential blowup in the number of recursions).

One way to avoid the inefficiency of knowledge extraction for SNARKs in the standard model is to consider knowledge extraction for SNARKs in oracle models, where one may hope to explicitly construct a highly-efficient knowledge extractor (rather than deduce its existence from a conservatively weak knowledge assumption). For example, in the random oracle model (ROM), the Micali construction [Mic00] yields a SNARK whose knowledge extractor is straightline [Val08; BCS16; CY24], i.e., the extractor only needs as input the query-answer trace of a single execution of the malicious SNARK prover. However, since recursive proof composition uses the underlying SNARK in a non-black-box way, it is not clear how to construct IVC and PCD from SNARKs in oracle models without first heuristically instantiating the oracles.

Remarkably, several works circumvent these difficulties by constructing *relativized SNARKs* in various oracle models, which are SNARKs that can prove computations that themselves involve queries to the oracle of the model. In particular, [CT10] constructs relativized SNARKs in the signed random oracle model (SROM); [CCS22] constructs relativized SNARKs in the low-degree random oracle model (LDROM); and [CCGOS23] constructs relativized SNARKs in the arithmetized random oracle model (AROM). Some of these models enable highly-efficient knowledge extractors, leading to significant improvements in the security reduction of recursive proof composition [CGSY23].

The aforementioned oracle models are substantially less efficient and less convenient to instantiate than the random oracle model, which is notably missing. In fact, the goal of achieving relativized SNARGs in alternate oracle models was motivated by the belief that relativized SNARGs in the ROM do not exist. While this belief is supported by circumstantial evidence (relativized PCPs/IOPs in the ROM do not exist [CL20]), there is no proof that confirms it. In this paper we study this question:

Do relativized SNARGs in the random oracle model exist?

1.1 Our results

In this work we prove that *relativized succinct arguments do not exist in the random oracle model (ROM)*. This resolves an open question left open in the line of work on relativized SNARGs, and stands in contrast to the fact that relativized SNARGs *are* possible in alternative oracle models of interest.

Below we informally recall the notions of relativized relations/languages and the relevant relativized complexity classes, and then we state our results in more detail. We denote by $\mathcal{O}$ the collection of random oracles: $\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}}$ where each $\mathcal{O}_\ell$ is the uniform distribution over functions $f : \{0,1\}^* \to \{0,1\}^\ell$.

Relativized relations and languages. A relation R is a set of instance-witness pairs $(\mathbb{x}, \mathbb{w})$, and a language is a set of instances $\mathbb{x}$. Since in this paper we study relativized complexity classes, we recall the relativized analogues of relations and languages. A *relativized relation* $R_{\mathcal{O}}$ is a collection $\{R_f\}_{f \in \mathcal{O}}$ where each R_f is a relation; similarly, a *relativized language* $L_{\mathcal{O}}$ is a collection $\{L_f\}_{f \in \mathcal{O}}$ where each L_f is a language.

Relativized complexity classes. We consider the relativized analogues of the complexity classes DTIME (deterministic computations bounded in time) and NTIME (nondeterministic computations bounded in time). Informally, for a given time-bound function $t \colon \mathbb{N} \to \mathbb{N}$, $\mathsf{DTIME}^{\mathcal{O}}[t]$ is the set of relativized languages decidable via a t-step deterministic Turing machine with query access to the random oracle (see Definition 3.11), and $\mathsf{NTIME}^{\mathcal{O}}[t]$ is the set of relativized relations decidable via a t-step nondeterministic Turing machine with query access to the random oracle (see Definition 3.9).

Similarly, we define a relativized complexity class for relativized non-interactive arguments, focusing on two efficiency metrics, the argument size and the query complexity of the argument verifier to the random oracle. Informally, $\mathsf{NARG}^{\mathcal{O}}[\mathsf{as}, \mathsf{vq}]$ is the set of relativized relations/languages that can be proved/verified via a relativized non-interactive argument with constant completeness error, constant soundness error against bounded-query adversaries, argument size as, and verifier query complexity vq to the random oracle (see Definition 3.14). All other efficiency parameters are polynomially bounded unless otherwise specified.

Separation for DTIME. In the standard model (with no oracles), SNARGs in the ROM enable verifying any t-step deterministic computation in time that is polylogarithmic in t. In sharp contrast, we prove that, in the ROM, relativized SNARGs for relativized deterministic computations do not exist.

Theorem 1 (informal). *Let $\mathcal{O}$ be the random oracle. For every time-constructible function $t \colon \mathbb{N} \to \mathbb{N}$,*

$$\mathsf{DTIME}^{\mathcal{O}}[t] \not\subseteq \mathsf{NARG}^{\mathcal{O}}[\mathsf{vq} = o(t)] \enspace .$$

The theorem considers an expressly weak notion of relativized SNARG: a relativized non-interactive argument for relativized deterministic computations where the "speedup" is that the argument verifier has query complexity to the random oracle $\mathsf{vq} = o(t)$, sublinear in the trivial query complexity of t (incurred by directly running the original computation); other efficiency parameters, including the argument verifier's running time and argument size, can be $\mathsf{poly}(t)$. The theorem says that, even when considering this minimal notion of succinctness, $\mathsf{DTIME}^{\mathcal{O}}[t]$ does not have relativized succinct non-interactive arguments.

Separation for NTIME. Next we consider the case of nondeterministic computations, where argument size is arguably the fundamental efficiency metric. In the standard model (with no oracles), SNARGs in the ROM enable verifying any t-step nondeterministic computation via arguments whose size is polylogarithmic in t. We prove that, in the ROM, relativized SNARGs for relativized nondeterministic computations do not exist.

Theorem 2 (informal). *Let $\mathcal{O}$ be the random oracle. For every time-constructible function $t \colon \mathbb{N} \to \mathbb{N}$,*

$$\mathsf{NTIME}^{\mathcal{O}}[t] \not\subseteq \mathsf{NARG}^{\mathcal{O}}[\mathsf{as} = o(t)] \enspace .$$

Similarly to Theorem 1, the theorem above considers an expressly weak notion of relativized SNARG: a relativized non-interactive argument for relativized nondeterministic computations where argument size is $\mathsf{as} = o(t)$, sublinear in t (the maximum size of a witness); other efficiency parameters, such as the the argument verifier's running time and query complexity to the random oracle, can be $\mathsf{poly}(t)$. The theorem says that, even when considering this minimal notion of succinctness, $\mathsf{NTIME}^{\mathcal{O}}[t]$ does not have relativized succinct non-interactive arguments.

The interactive case. So far we have discussed (and ruled out) relativized SNARGs in the ROM. In fact, our techniques extend to establish similarly strong impossibility results for relativized succinct *interactive* arguments (the argument prover and argument verifier may interact across multiple rounds). Informally,

$\mathsf{IARG}^{\mathcal{O}}[\mathsf{pc}, \mathsf{vq}]$ is the set of relativized languages that can be proved/verified by an interactive argument with constant completeness error, constant soundness error against bounded-query adversaries, verifier query complexity vq, and prover communication complexity pc (see Definition 7.4).

Lemma 1 (informal). *Let $\mathcal{O}$ be the random oracle. For every time-constructible function $t\colon \mathbb{N} \to \mathbb{N}$,*

$$\mathsf{DTIME}^{\mathcal{O}}[t] \not\subseteq \mathsf{IARG}^{\mathcal{O}}[\mathsf{vq} = o(t)] \quad \textit{and} \quad \mathsf{NTIME}^{\mathcal{O}}[t] \not\subseteq \mathsf{IARG}^{\mathcal{O}}[\mathsf{pc} = o(t)] \ .$$

Computational soundness. All negative results that we discussed so far hold even if the (non-interactive or interactive) argument is merely computationally sound: the soundness notion is relaxed to apply only for computationally bounded adversaries (which, in particular, are query bounded). This extension significantly strengthens our impossibility results, putting them in sharp contrast with positive results that achieve relativized SNARGs with computational soundness in other oracle models [CT10; CCS22; CCGOS23].

Relativized IVC and PCD in the ROM. Known constructions of IVC and PCD in the standard model (without oracles) rely on knowledge assumptions or heuristics. On the other hand, IVC and PCD can be achieved, in various oracle models, based on falsifiable assumptions (by relying on suitable relativized SNARKs in those models). The most desirable oracle model remains the ROM; however, the belief is that, regrettably, IVC and PCD in the ROM cannot be achieved based on falsifiable assumptions. A partial negative result supporting this belief is that zero-knowledge IVC in the ROM does not exist (provided the existence of a suitable commitment scheme) [HN23]. The results in this paper provide additional results supporting this belief: our results imply that *relativized* IVC and PCD in the ROM do not exist. This is because the existence of relativized IVC (in particular, PCD) in the ROM would imply, via the bootstrapping approach in [BCCT13], the existence of a relativized SNARG in the ROM (which is ruled out by our results). Ruling out the existence of standard (non-relativized) IVC and PCD in the ROM remains an open problem.

1.2 Related work

Relativized SNARGs in other oracle models. Several works construct relativized SNARKs (SNARGs of knowledge) in oracle models of interest. [CT10] constructs relativized SNARKs in the signed random oracle model (SROM), a model that combines a signature scheme with the random oracle model. [CCS22] constructs relativized SNARKs in the low-degree random oracle model (LDROM), an oracle model that considers random low-degree extensions of random oracles. [CCGOS23] constructs relativized SNARKs in the arithmetized random oracle model (AROM), an idealization of capabilities associated to the arithmetization of a hash function. In all of these cases, the relativized SNARKs directly imply corresponding black-box constructions of PCD (proof-carrying data) in the respective oracle model; if the relativized SNARK is straightline extractable then the security of PCD is particularly efficient [CGSY23].

SNARGs in the ROM. Standard (non-relativized) SNARGs in the ROM are obtained from probabilistic proofs [Mic00; BCS16; CY24]; the use of probabilistic proofs is, in a precise sense, inherent [CY20]. A SNARG in the ROM can be straightforwardly made a "relativized NARK" in the ROM by sacrificing succinctness (which is consistent with the impossibility results in this paper); this observation can be used towards studying the concrete security of hash-based PCD constructions [CGSY23].

Probabilistic proofs do not relativize. Several works establish impossibility results for relativized probabilistic proofs. [For94] constructs a function f such that, for every $k \in \mathbb{N}$, $\mathsf{NP}^f \not\subseteq \mathsf{PCP}^f[\mathbb{q} = n^k, \mathbb{vt} = \mathsf{poly}(n)]$. [Cha+94] shows that $\Pr_{f \in \mathcal{O}}\left[\mathsf{IP}^f \not\subseteq \mathsf{PSPACE}^f\right] = 1$. [CL20] shows impossibility results for relativized PCPs and IOPs. The techniques in this paper build on this line of work; we elaborate on this in Section 2.

Relativization and other barriers. A line of work studies the limitations of certain proof techniques for establishing results in structural complexity theory [Dek69; BGS75; Lis86; Hel86; AIV92; For94; Cha+94; AW09; IKK09; AB18]. For example, the relativization barrier refers to the limitations of proof techniques that are independent of which "relativized world" they live in. As discussed in Section 2.3, the techniques used in this work have their origins in works that study the relativization barrier. However, the goal of this work is different: we show that relativized succinct arguments in the ROM do not exist, regardless of any underlying techniques that one may attempt to use to construct them.

2 Techniques

We outline the main ideas underlying our results. Later sections contain the technical details.

2.1 Review: relativized SNARGs

We informally describe relativized SNARGs in the ROM to facilitate discussions in this technical overview; formal definitions are provided in Section 3.4. A *relativized SNARG in the ROM* is a SNARG for relativized relations/languages in the ROM. Namely, we say that $(\mathcal{P}, \mathcal{V})$ is a SNARG for a relativized relation $R_{\mathcal{O}}$ if it satisfies the completeness and soundness properties below. (Since in this work we prove negative results, both properties are deliberately weak: we allow large completeness and soundness errors.)

- *Completeness*: For every security parameter $\lambda \in \mathbb{N}$ and every adversary $\mathcal{A}$,

$$\Pr\left[\begin{array}{c} (\mathbb{x}, \mathbb{w}) \in R_f \\ \wedge\, \mathcal{V}^f(\mathbb{x}, \pi) = 0 \end{array} \,\middle|\, \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ \mathbb{x} \leftarrow \mathcal{A}^f \\ \pi \leftarrow \mathcal{P}^f(\mathbb{x}, \mathbb{w}) \end{array}\right] \leq \frac{1}{3} \ .$$

- *Soundness*: For every security parameter $\lambda \in \mathbb{N}$ and every $\mathsf{poly}(\lambda)$-query adversary $\widetilde{\mathcal{P}}$,

$$\Pr\left[\begin{array}{c} \mathbb{x} \notin L(R_f) \\ \wedge\, \mathcal{V}^f(\mathbb{x}, \pi) = 1 \end{array} \,\middle|\, \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (\mathbb{x}, \pi) \leftarrow \widetilde{\mathcal{P}}^f \end{array}\right] \leq \frac{1}{3} \ .$$

The case of a SNARG for a relativized language $L_{\mathcal{O}}$ is the special case where there are no witnesses.[1]

We shall care about the following efficiency measures: pq denotes the query complexity of the (honest) prover $\mathcal{P}$; vq denotes the query complexity of the verifier $\mathcal{V}$; and $\mathsf{as} := |\pi|$ denotes the argument size.

In this technical overview, we suppress the security parameter λ for simplicity.

2.2 Why not reduce relativized arguments to relativized probabilistic proofs?

All known SNARG constructions in the ROM are based on probabilistic proofs and, in fact, any non-trivial SNARG in the ROM implies a non-trivial probabilistic proof [CY20]. This suggests a natural approach: can we reduce the existence of non-trivial *relativized* SNARGs in the ROM to non-trivial *relativized* probabilistic proofs in the ROM? This may suffice for our goal because [CL20] establishes strong impossibility results for relativized probabilistic proofs in the ROM.

We explain how this reasonable approach yields some progress but does not suffice for our results.

For concreteness, we recall the limitations for relativized PCPs. Let $\mathsf{PCP}^{\mathcal{O}}[\mathbb{q}, \mathbb{vt}]$ be the set of relativized languages $L_{\mathcal{O}}$ for which there exists a PCP verifier $\mathbb{V}$ (a probabilistic Turing machine) with oracle access to the random oracle that runs in time $\mathbb{vt}$ and makes $\mathbb{q}$ queries to the given PCP string such that

$$\Pr_{f \in \mathcal{O}}\left[\mathbb{V}^f \text{ is a PCP verifier for } R_f\right] = 1 \ .$$

[CL20] shows that, for every time-constructible function $t \colon \mathbb{N} \to \mathbb{N}$, $\mathsf{NTIME}^{\mathcal{O}}[t]$ and $\mathsf{DTIME}^{\mathcal{O}}[t]$ do not have non-trivial relativized PCPs in the ROM:

$$\mathsf{DTIME}^{\mathcal{O}}[t] \not\subseteq \mathsf{PCP}^{\mathcal{O}}[\mathbb{vt} = o(t)] \quad \text{and} \quad \mathsf{NTIME}^{\mathcal{O}}[t] \not\subseteq \mathsf{PCP}^{\mathcal{O}}[\mathbb{q} = o(t)] \ .$$

[1] Explicitly: in completeness, the honest prover receives as input only the instance $\mathbb{x}$ and the condition $\mathbb{x} \in L_f$ replaces the condition $(\mathbb{x}, \mathbb{w}) \in R_f$; and, in soundness, the condition $\mathbb{x} \notin L_f$ replaces the condition $\mathbb{x} \notin L(R_f)$.

Indeed, for deterministic computation they consider the expressly weak goal of PCP verifier time $\mathbb{vt} = o(t)$ (slightly better than the trivial decision time of t) and for nondeterministic computations they consider the expressly weak goal of PCP verifier query complexity $\mathbb{q} = o(t)$ (slightly better than the trivial witness size).

In fact, their results straightforwardly extend to the case of interactive oracle proofs (IOPs), which are a multi-round analogue of PCPs:

$$\mathsf{DTIME}^{\mathcal{O}}[t] \not\subseteq \mathsf{IOP}^{\mathcal{O}}[\mathbb{vt} = o(t)] \quad \text{and} \quad \mathsf{NTIME}^{\mathcal{O}}[t] \not\subseteq \mathsf{IOP}^{\mathcal{O}}[\mathbb{q} = o(t)] \ . \tag{1}$$

Ideally, we would like to show that a non-trivial relativized SNARG in the ROM contradicts the above impossibility results. Towards this end, we recall that [CY20] provides a transformation from any SNARG in the ROM to a non-trivial IOP (in the standard model without oracles). Given a non-interactive argument in the ROM with constant completeness error, constant soundness error (against bounded-query adversaries), argument size as, prover query complexity to the random oracle pq, verifier query complexity to the random oracle vq, and verifier running time vt, the transformation outputs an IOP with constant completeness error, constant soundness error, verifier running time $O(\mathsf{vt} + \mathsf{vq} \cdot \mathsf{pq})$, and verifier query complexity $O(\mathsf{as} + \mathsf{vq})$ (to the IOP strings).[2]

We observe that the transformation in [CY20] can be adapted to the relativized setting with essentially no losses in parameters: the input to the transformation is a relativized SNARG in the ROM, and the output is a relativized IOP in the ROM. Thus, we get that if a relativized language $L_{\mathcal{O}}$ is in $\mathsf{NARG}^{\mathcal{O}}[\mathsf{pq}, \mathsf{as}, \mathsf{vq}, \mathsf{vt}]$ then $L_{\mathcal{O}}$ is in $\mathsf{IOP}^{\mathcal{O}}[\mathbb{q} = \mathsf{as} + \mathsf{vq}, \mathbb{vt} = \mathsf{vt} + \mathsf{vq} \cdot \mathsf{pq}]$. We can combine this transformation with the impossibility results for relativized IOPs in Equation (1) as follows.

Consider a relativized non-interactive argument with $\mathsf{vt} = o(t)$ and $\mathsf{vq} \cdot \mathsf{pq} = o(t)$. The transformation produces a relativized IOP with verifier running time $\mathbb{vt} = \mathsf{vt} + \mathsf{vq} \cdot \mathsf{pq} = o(t)$. Then, by Equation (1), we conclude that

$$\mathsf{DTIME}^{\mathcal{O}}[t] \not\subseteq \mathsf{NARG}^{\mathcal{O}}[\mathsf{vt} = o(t), \mathsf{vq} \cdot \mathsf{pq} = o(t)] \ .$$

Similarly, the transformation applied to a relativized non-interactive argument with $\mathsf{vq} = o(t)$ and $\mathsf{as} = o(t)$ yields a relativized IOP with verifier query complexity $\mathbb{q} = \mathsf{as} + \mathsf{vq} = o(t)$. Then, by Equation (1), we conclude that

$$\mathsf{NTIME}^{\mathcal{O}}[t] \not\subseteq \mathsf{NARG}^{\mathcal{O}}[\mathsf{as} = o(t), \mathsf{vq} = o(t)] \ .$$

Both of these statements are strictly weaker than our Theorem 1 and Theorem 2: the DTIME result imposes considerable unnecessary constraints on the verifier running time vt and (unrealistically) the honest prover query complexity pq; and the NTIME result imposes an unnecessary constraint on the verifier query complexity vq. Whether the transformation of [CY20] can be further improved (and adapted to the relativized setting) remains an open problem; in this paper we take a different, more direct, approach to prove our results.

2.3 Prior techniques for oracle separations

A line of works show the impossibility of probabilistic proofs in relativized worlds. [For94] uses diagonalization to construct a function f such that, for every $k \in \mathbb{N}$, $\mathsf{NP}^f \not\subseteq \mathsf{PCP}^f[\mathbb{q} = n^k, \mathbb{vt} = \mathsf{poly}(n)]$. [Cha+94] shows that $\mathsf{IP}^{\mathcal{O}} \not\subseteq \mathsf{PSPACE}^{\mathcal{O}}$, more precisely, that $\Pr_{f \in \mathcal{O}}\left[\mathsf{IP}^f \not\subseteq \mathsf{PSPACE}^f\right] = 1$. [CL20] shows the impossibility results for relativized PCPs and IOPs discussed in Section 2.2. The techniques used across these works are interrelated. As we shall build on these techniques, we find it helpful to review the ideas underlying, e.g., the separation $\mathsf{DTIME}^{\mathcal{O}}[t] \not\subseteq \mathsf{PCP}^{\mathcal{O}}[\mathbb{vt} = o(t)]$ in [CL20].

[2] The statement also extends to work for any given completeness error and soundness error.

The proof consists of identifying a relativized language $L^\star_\mathcal{O}$ that is in $\mathsf{DTIME}^\mathcal{O}[t]$ (easy to prove) but not in $\mathsf{PCP}^\mathcal{O}[\mathbb{vt} = o(t)]$ (the actual challenge).

Consider two oracles $f_1\colon \{0,1\}^* \to \{0,1\}^*$ and $f_2\colon \{0,1\}^* \to \{0,1\}^*$ that differ in a random location $q \in \{0,1\}^t$. No $o(t)$-time algorithm can distinguish f_1 and f_2 with non-negligible probability. More generally, if f_1 and f_2 differ on a small random set $Q \subseteq \{0,1\}^t$, no $o(t)$-time algorithm can tell them apart. Below we use $\mathsf{dist}(f_1, f_2)$ to denote the (possibly infinite) number of $x \in \{0,1\}^*$ such that $f_1(x) \neq f_2(x)$.

Using the intuition above, the relativized language $L^\star_\mathcal{O}$ is designed such that there exists a special instance $\mathbb{x}$ whose membership in $L^\star_\mathcal{O}$ depends on the answers to a set $Q_\mathbb{x}$ of t different queries to the oracle function. There is a set of oracle functions $F \subseteq \{f \in \mathcal{O} : \mathbb{x} \in L^\star_f\}$ with non-zero measure such that for every $f \in F$ there is another function $f' \in \mathcal{O}$ for which $\mathbb{x} \notin L^\star_{f'}$ and $\mathsf{dist}(f, f') \ll t$.

By the completeness of the PCP, since $\mathbb{x} \in L^\star_f$, there exists a PCP proof Π_f such that $\mathbb{V}^{f,\Pi_f}(\mathbb{x}) = 1$ with high probability. On the other hand, since the PCP verifier $\mathbb{V}$ makes $o(t)$ queries to the random oracle ($\mathbb{V}$ runs in time $o(t)$), with high probability $\mathbb{V}$ cannot distinguish f and f'. We deduce that $\mathbb{V}^{f',\Pi_f}(\mathbb{x})$ also accepts with high probability. However, this contradicts the soundness of the PCP since $\mathbb{x} \notin L^\star_{f'}$. Since the measure of the set F is non-zero, we deduce that $L^\star_\mathcal{O} \notin \mathsf{PCP}^\mathcal{O}[\mathbb{vt} = o(t)]$, as desired.

Informally, we directly prove that relativized SNARGs in the ROM do not exist, by building on the above ideas. This requires overcoming several challenges, as we discuss in Section 2.4 for the case $\mathsf{DTIME}^\mathcal{O}[t]$ and then in Section 2.5 for the (harder) case of $\mathsf{NTIME}^\mathcal{O}[t]$.

2.4 Separation between DTIME and NARG

The contradiction in Section 2.3 is obtained by finding a set of functions with non-zero measure for which PCP soundness is not satisfied. From this starting point we encounter two challenges to prove Theorem 1.

Challenge 1: soundness is average-case. The definition of a PCP in the ROM for relativized languages requires that, with probability 1 over the choice of the random oracle $f \in \mathcal{O}$, the PCP verifier $\mathbb{V}$ decides L_f with constant completeness error and constant soundness error. On the other hand, SNARGs are defined so that the completeness and soundness errors are relative to a random choice of oracle $f \in \mathcal{O}$ (see Section 2.1).[3] Unfortunately, the proof sketch in Section 2.3 fails to work because it focuses on oracles that lie in a small subset $F \subseteq \mathcal{O}$ of measure roughly 2^{-t}, which does not suffice to contradict SNARG soundness, unless the completeness and the soundness error of the SNARG are exponentially close to each other.[4]

Challenge 2: soundness is against query-bounded adversaries. The security guarantee of PCPs and SNARGs are qualitatively different: PCPs are information-theoretically sound, while SNARGs are sound against bounded-query adversaries.

Recall that in Section 2.3, for every f in the hard set F, one can construct another function f' such that $\mathbb{x} \notin L^\star_{f'}$ and $\mathsf{dist}(f, f') \ll t$. Moreover, there exists a PCP string Π_f such that the PCP verifier $\mathbb{V}^{f',\Pi_f}(\mathbb{x})$ accepts with high probability over the PCP randomness. In particular, by completeness of the PCP, the PCP string Π_f can be generated by the honest PCP prover $\mathbb{P}$ given oracle access to f (since $\mathbb{x} \in L^\star_f$).

However, in the context of SNARGs, this approach cannot be directly applied. Let π_f be the argument string generated by the honest argument prover $\mathcal{P}^f(\mathbb{x})$. First, we cannot deduce from the SNARG completeness that $\mathcal{V}^f(\mathbb{x}, \pi_f)$ accepts with high probability over the private randomness of $\mathcal{V}$ (because of

[3] This is consistent with all prior work on relativized SNARGs, and also with the definitions of completeness and soundness for standard (non-relativized) SNARGs. In fact, requiring soundness to hold with probability 1 over the choice of random oracle would be too strong (all known non-relativized SNARGs in the ROM [Mic00; BCS16; CY24] do *not* satisfy such a strong notion).

[4] One would need $\frac{\beta}{1-\alpha} < 2^{-t}$ where α and β are the completeness error and the soundness error of the non-interactive argument. This is a restrictive and extremal parameter regime.

Challenge 1). Moreover, even if we could conclude that $\mathcal{V}^{f'}(\mathbb{x}, \pi_f)$ accepts with high probability over the private randomness of $\mathcal{V}$, this does *not* contradict the SNARG soundness unless π_f can be generated by a query-bounded adversary with oracle access to f' (rather than f).

Solution 1. Some prior work proves stronger statements than needed for random oracle separations. For example, [CL20] shows an *almost-everywhere separation* between $\mathsf{DTIME}^{\mathcal{O}}$ and $\mathsf{PCP}^{\mathcal{O}}$:

$$\Pr_{f \in \mathcal{O}}\left[\mathsf{DTIME}^f \not\subseteq \mathsf{PCP}^f[\mathbb{vt} = o(t)]\right] = 1 \ .$$

We cannot hope to show such a strong statement for relativized arguments. Nevertheless, this statement provides a better starting point for us to reason about the average behavior over $f \in \mathcal{O}$. Inspired by the proof of the almost-everywhere separation, we observe that for every $f \in \mathcal{O}$ there is a large subset $Q_f \subseteq Q_{\mathbb{x}}$ ($|Q_f| \geq t - o(t)$) where no query-bounded verifier $\mathcal{V}^f$ queries any location in Q_f with high probability. Hence, for a randomly sampled f such that $\mathbb{x} \in L_f^\star$, one can construct f' where $\mathbb{x} \notin L_{f'}^\star$ by flipping the answers to a random query in $Q_{\mathbb{x}}$ without being detected by any query-bounded verifier $\mathcal{V}$. More formally,

$$\Pr\left[\mathcal{V} \text{ distinguish } f \text{ and } f' \;\middle|\; \begin{array}{c} f \leftarrow \mathcal{O} \\ q \leftarrow Q_{\mathbb{x}} \end{array}\right] = \Pr\left[q \in Q_f \;\middle|\; \begin{array}{c} f \leftarrow \mathcal{O} \\ q \leftarrow Q_{\mathbb{x}} \end{array}\right] \geq \frac{t - o(t)}{t} \ .$$

This allows us to adapt ideas in Section 2.3 and apply them to every $f \in \mathcal{O}$ instead of to only a small set F.

Solution 2. We construct a query-efficient argument adversary $\widetilde{\mathcal{P}}^{f'}$ that computes the argument string π_f output by the honest argument prover $\mathcal{P}^f(\mathbb{x})$ (note the different oracles). The relativized language $L_{\mathcal{O}}^\star$ is designed so that, for every $f \in F$, there is a special query $q_f \in Q_{\mathbb{x}}$ where one can define f' such that $\mathbb{x} \notin L_{f'}^\star$ as follows:

- for every $q \neq q_f$, $f'(q) := f(q)$;
- $f'(q_f)$ is the bitwise complement of $f(q_f)$.

In fact, there is a one-to-one correspondence between f and f'. The adversary $\widetilde{\mathcal{P}}^{f'}(\mathbb{x})$ is defined as follows.

> $\widetilde{\mathcal{P}}^{f'}(\mathbb{x})$: Simulate the honest prover $\mathcal{P}(\mathbb{x})$ by using $f'(q)$ to answer queries $q \neq q_f$ and using the bitwise complement of $f'(q_f)$ to answer the query q_f; this leads to an output π. Output π.

Clearly, $\widetilde{\mathcal{P}}^{f'}(\mathbb{x}) = \mathcal{P}^f(\mathbb{x})$. Moreover, the number of queries made by $\widetilde{\mathcal{P}}$ to f' equals the number of queries made by $\mathcal{P}$ to f. Since $\mathcal{P}$ is query efficient so is $\widetilde{\mathcal{P}}$, and we get a contradiction to the SNARG soundness.

2.5 Separation between NTIME and NARG

Theorem 1 (separation between $\mathsf{DTIME}^{\mathcal{O}}$ and $\mathsf{NARG}^{\mathcal{O}}$) rules out relativized SNARGs where the verifier has query complexity $\mathsf{vq} = o(t)$ to the random oracle. Instead, Theorem 2 (separation between $\mathsf{NTIME}^{\mathcal{O}}$ and $\mathsf{NARG}^{\mathcal{O}}$) rules out relativized SNARGs with argument size $\mathsf{as} = o(t)$, *even if* the verifier makes $\mathsf{vq} = \mathsf{poly}(t)$ queries to the random oracle. The relativized language $L_{\mathcal{O}}^\star$ we discussed in Section 2.4 can be decided by an argument verifier with $\mathsf{poly}(t)$ queries, without any help from the argument prover. Hence, we must consider languages/relations that cannot be decided with $\mathsf{poly}(t)$ queries.

The relativized relation $R_{\mathcal{O}}^\star = \{R_f^\star\}_{f \in \mathcal{O}}$ we use to prove Theorem 2 has the following form:

$$R_f^\star := \left\{(\mathbb{x}, \mathbb{w}) \in \{0,1\}^n \times \{0,1\}^{t(n)} : \forall\, q \in S_{\mathbb{x},\mathbb{w}},\ \text{the first bit of } f(q) \text{ is } 0\right\} \ ,$$

where $S_{\mathbb{x},\mathbb{w}}$ is a set of $O(t)$ queries and, for every $\mathbb{w}' \neq \mathbb{w}$, $S_{\mathbb{x},\mathbb{w}} \cap S_{\mathbb{x},\mathbb{w}'} = \emptyset$. The relativized relation $R^{\star}_{\mathcal{O}}$ can be (straightforwardly) decided by a nondeterministic Turing machine that runs in time $O(t)$. On the other hand, it is unclear how a $\mathsf{poly}(t)$-query argument verifier $\mathcal{V}$, on input $\mathbb{x}$, can decide whether there exists a witness $\mathbb{w}$ such that $(\mathbb{x}, \mathbb{w}) \in R^{\star}_{\mathcal{O}}$; finding the witness alone would take much more than $\mathsf{poly}(t)$ queries.

Recall that Theorem 1 relied on the fact that an argument verifier $\mathcal{V}$ with small query complexity cannot distinguish f_1 and f_2 that differ at few random locations. While now the argument verifier $\mathcal{V}$ can make $\mathsf{poly}(t)$ queries, we can nevertheless deduce an upper bound on the maximum total number of distinct queries that $\mathcal{V}$ can make across all possible argument strings. For every random oracle $f \in \mathcal{O}$ and argument string π, let $N(\mathcal{V}^f(\mathbb{x}, \pi))$ be the number of distinct queries $\mathcal{V}(\mathbb{x}, \pi)$ makes to f. Then, for every $f \in \mathcal{O}$,

$$\sum_{\pi \in \{0,1\}^{o(t)}} N(\mathcal{V}^f(\mathbb{x}, \pi)) \leq \sum_{\pi \in \{0,1\}^{o(t)}} \mathsf{poly}(t) = 2^{o(t)} \cdot \mathsf{poly}(t) = 2^{o(t)} \ .$$

Namely, $\mathcal{V}^f(\mathbb{x}, \cdot)$ makes at most $2^{o(t)}$ distinct queries to the random oracle f across all possible argument strings $\pi \in \{0,1\}^{o(t)}$. We do not expect an algorithm to decide the membership of $\mathbb{x}$ in the relativized language $L(R^{\star}_{\mathcal{O}})$ with $2^{o(t)}$ queries.

However, the ideas underlying the proof of Theorem 1, in which random oracles are partitioned via a one-to-one correspondence, do not seem to extend: the relativized relation $R_{\mathcal{O}} \in \mathsf{NTIME}^{\mathcal{O}}[t]$ does not satisfy this property (nor do we know of other hard relations that do). Given $f \in \mathcal{O}$ where $\mathbb{x} \in L(R^{\star}_f)$, one has to find every $\mathbb{w} \in \{0,1\}^t$ such that $(\mathbb{x}, \mathbb{w}) \in R^{\star}_f$ in order to construct f' with $\mathbb{x} \notin L(R^{\star}_{f'})$. On the other hand, given $f \in \mathcal{O}$ and $\mathbb{x} \notin L(R^{\star}_f)$, one can easily construct f' such that $\mathbb{x} \in L(R^{\star}_{f'})$: choose an arbitrary $\mathbb{w} \in \{0,1\}^t$ and, for every $q \in S_{\mathbb{x},\mathbb{w}}$, set the first bit of $f'(q)$ to 0. However, this construction is "destructive": we cannot revert f' back to f. In other words, it is not possible to simulate f with f' as in Section 2.4.

The key idea here is that although we cannot hope to have the one-to-one correspondence, it is sufficient to obtain a one-to-M correspondence for some fixed constant M. In particular, consider an arbitrary $f \in \mathcal{O}$ such that $\mathbb{x} \notin L(R^{\star}_f)$. For every possible witness $\mathbb{w} \in \{0,1\}^t$, we can construct $f' \in \mathcal{O}$ where $(\mathbb{x}, \mathbb{w}) \in R^{\star}_{f'}$ by setting the first bit of $f'(q)$ to 0 for every $q \in S_{\mathbb{x},\mathbb{w}}$. Hence, every f with $\mathbb{x} \notin L(R^{\star}_f)$ corresponds to $M = 2^t$ different f' with $\mathbb{x} \in L(R^{\star}_{f'})$. The rest of the argument follows similarly as in Section 2.4 with a careful counting analysis; we refer the readers to Section 5 for more details.

Remark 1. *Since* $\mathsf{DTIME}^{\mathcal{O}}[t] \subseteq \mathsf{NTIME}^{\mathcal{O}}[t]$, *Theorem 1 implies the following statement*

$$\mathsf{NTIME}^{\mathcal{O}}[t] \not\subseteq \mathsf{NARG}^{\mathcal{O}}[\mathsf{vq} = o(t)] \ . \tag{2}$$

In other words, $\mathsf{NTIME}^{\mathcal{O}}[t]$ *does not have relativized non-interactive arguments where the argument verifier makes* $\mathsf{vq} = o(t)$ *queries to the random oracle, regardless of the argument size.*

In Section 6 we provide a "direct" proof of Equation (2)*, incomparable to the proof of Theorem 1. We prove that the relativized relation* $R^{\star}_{\mathcal{O}}$ *above (used to prove Theorem 2) is* also *not in* $\mathsf{NARG}^{\mathcal{O}}[\mathsf{vq} = o(t)]$.

We outline the proof idea. Similarly to the proof of Theorem 1, if the argument verifier $\mathcal{V}$ *makes* $\mathsf{vq} = o(t)$ *queries to the random oracle, then it cannot distinguish between* f *and* f' *that differ at few random locations. However, the challenge is to obtain the one-to-one correspondence.*

Given $f \in \mathcal{O}$ *with* $(\mathbb{x}, \mathbb{w}) \in R^{\star}_f$*, it is easy to construct* f' *such that* $(\mathbb{x}, \mathbb{w}) \notin R^{\star}_{f'}$*: set the first bit of* $f'(q)$ *to* 1 *for all* $q \in S_{\mathbb{x},\mathbb{w}}$. *However, it still might be the case that* $\mathbb{x} \in L(R^{\star}_{f'})$ *because there could be another witness* $\mathbb{w}' \neq \mathbb{w}$ *such that* $(\mathbb{x}, \mathbb{w}') \in R^{\star}_{f'}$. *We resolve this issue by considering the following set* U_1 *of oracle instances:*

$$U_1 \coloneqq \left\{ f \in \mathcal{O} : \exists!\, \mathbb{w} \text{ s.t. } (\mathbb{x}, \mathbb{w}) \in R^{\star}_f \right\} \ .$$

For every f in the above set, we can construct f' by flipping the first bit of $f(q)$ for an arbitrary $q \in S_{\mathbb{x},\mathbb{w}}$ to ensure that $\mathbb{x} \notin R^{\star}_{f'}$. Moreover, one can easily simulate f given oracle access to f'. Since the measure of the set U_1 is a constant, we can adapt the proof for Theorem 1 and get the desired result.

We stress that the above approach does not *work when proving Theorem 2, as an argument verifier with* $\mathsf{poly}(t)$ *queries can distinguish $f \in U_1$ and its corresponding f' with high probability.*

2.6 The case of relativized interactive arguments

Our discussions so far have focused on (proving the impossibility of) relativized SNARGs in the ROM. Indeed, relativized SNARGs directly lead to PCD and they have been achieved in various oracle models beyond the ROM [CT10; CCS22; CCGOS23; CGSY23]. However, from the perspective of an impossibility result, we find it natural to additionally ask: *Do relativized succinct arguments in the ROM exist, if we allow the succinct argument to be interactive?* We explain how, with this relaxation, the answer remains negative.

Why not unroll the interactive argument? A first attempt to prove the impossibility of relativized succinct interactive arguments may be to reduce the interactive case to the non-interactive case.

Indeed, this *is* possible in the case of probabilistic proofs: [CL20] establishes the impossibility of relativized PCPs in the ROM, and subsequently generically reduces relativized IOPs (the multi-round generalization of PCPs) to the case of relativized PCPs. This reduction is rather straightforward: any IOP (relativized or not) can be "unrolled" into a (much longer!) corresponding PCP while, crucially, leaving the efficiency of the verifier intact (the verifier computation and queries remain unaffected). Since the results in [CL20] depend on the efficiency of the verifier, the impossibility of relativized IOPs in the ROM follows from the impossibility of relativized PCPs in the ROM.

We can similarly "unroll" an interactive argument to obtain a corresponding non-interactive protocol that can be viewed as a relaxation of a non-interactive argument where the argument verifier has query access to the argument string; in other words, a *probabilistically-checkable argument* [KR09] in the ROM that is secure against query-bounded adversaries. Given a relativized interactive argument $\mathsf{IARG} = (\mathcal{P}_1, \mathcal{V}_1)$ for an relativized relation $R_{\mathcal{O}}$, define a relativized probabilistically-checkable argument $\mathsf{PCA} = (\mathcal{P}_2, \mathcal{V}_2)$ for $R_{\mathcal{O}}$.

- $\mathcal{P}_2^f(\mathbb{x}, \mathbb{w})$:
  1. For every possible verifier randomness $\rho \in \{0,1\}^{\mathsf{vr}}$: simulate the interaction $\langle \mathcal{P}_1^f(\mathbb{x}, \mathbb{w}), \mathcal{V}^f(\mathbb{x}, \rho) \rangle$ and let $(b_{\rho,i})_{i \in [\mathsf{k}]}$ be the list of messages sent by $\mathcal{P}_1^f(1^\lambda, \mathbb{x}, \mathbb{w})$ during the interaction.
  2. Output the argument string $\pi := (((b_{\rho,i})_{i \in [\mathsf{k}]}))_{\rho \in \{0,1\}^{\mathsf{vr}}}$.
- $\mathcal{V}_2^{f,\pi}(\mathbb{x})$:
  1. Sample verifier randomness $\rho \leftarrow \{0,1\}^{\mathsf{vr}}$.
  2. Query π at ρ to obtain $(b_i)_{i \in [\mathsf{k}]} := \pi[\rho]$.
  3. Output the decision bit $\mathsf{d} := \mathcal{V}_1^f(\mathbb{x}, \rho, (b_i)_{i \in [\mathsf{k}]})$.

While the argument string π is huge, the argument verifier only reads a small portion of π corresponding to a single interaction and, moreover, the verifier query complexity to the random oracle is unaffected by the unrolling. In particular, if $\mathsf{IARG} = (\mathcal{P}_1, \mathcal{V}_1)$ is succinct then $\mathsf{PCA} = (\mathcal{P}_2, \mathcal{V}_2)$ is "non-trivial".

Can we hope to prove impossibility results for interactive arguments (as in our Lemma 1) by: (a) extending our impossibility results for non-interactive arguments to the case of probabilistically-checkable arguments, and then (b) relying on the above reduction? The extension can be carried out (the techniques directly extend); however, there a problem arises in the second step. As discussed in Section 2.4, in the context of relativized arguments, it is important that the honest argument prover queries the random oracle at most polynomially many times, otherwise the soundness guarantee does not yield any contradiction (it only applies to query-bounded adversaries). In the transformation above, the query complexity of the honest argument prover increases by a multiplicative factor of 2^{vr}. Overall, this unrolling approach seems to fail.

Our approach. We take a direct approach to proving Lemma 1. We observe that the ideas that we sketched in Sections 2.4 and 2.5 can be "lifted" to directly work for interactive arguments, where the prover-to-verifier communication complexity pc replaces the argument size as (and the argument verifier query complexity to the random oracle remains the same notion). For example, we rely on the fact that the argument verifier does not have enough queries to distinguish "close but opposite" oracle functions. In the context of non-interactive arguments, the number of distinct queries an argument verifier can make for a give instance $\mathbb{x}$ depends only on its query complexity and the argument size. Similarly, in the context of interactive arguments, the number of distinct queries the verifier can make for a given instance $\mathbb{x}$ depends only on its query complexity and the prover-to-verifier communication complexity. Hence, we can extend the proofs for non-interactive arguments directly to work for interactive arguments.

In fact, our impossibility results for relativized interactive arguments imply our results for relativized non-interactive arguments as a special case. Nevertheless, we find it helpful to keep the technical overview and technical sections focused on the notationally simpler case of non-interactive arguments. Hence, the technical details underlying our results for non-interactive arguments are in Sections 4 to 6 and, separately, we explain how to extend them to work for interactive arguments in Section 7.

2.7 The random oracle model with cryptography

We have so far discussed relativized NARGs (and IARGs) in the "pure" ROM, namely, where adversaries are query bounded but can be computationally unbounded. Here we discuss a relaxation, where the soundness notion of a relativized SNARG applies only for computationally bounded adversaries (which, in particular, are query bounded). Informally, $(\mathcal{P}, \mathcal{V})$ is a *computationally-sound* SNARG for a relativized relation $R_{\mathcal{O}}$ if it satisfies the following relaxed notion of soundness:

- *Computational soundness*: For every security parameter $\lambda \in \mathbb{N}$ and every $\mathsf{poly}(\lambda)$-size adversary $\widetilde{\mathcal{P}}$,

$$\Pr\left[\begin{array}{l} \mathbb{x} \notin L(R_f) \\ \wedge\, \mathcal{V}^f(\mathbb{x}, \pi) = 1 \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (\mathbb{x}, \pi) \leftarrow \widetilde{\mathcal{P}}^f \end{array}\right] \leq \frac{1}{3} \ .$$

Such a relaxation is *essential* for all known constructions of relativized SNARGs in other oracle models: [CT10; CCS22; CCGOS23] all rely on some computational hardness assumption to achieve relativized SNARGs in their respective oracle models. This raises the question: *can we rule out the existence of relativized SNARGs (and IARGs) that only satisfy computational soundness?*

We show that the proof ideas sketched so far extend to rule out relativized SNARGs (and IARGs) with computational soundness. Intuitively, this is because the argument adversaries outlined in Sections 2.4 and 2.5 preserve the query complexity *and the running time* of the honest argument prover. Therefore we can still invoke the computational soundness property to reach a contradiction. The formal definitions, theorem statements, and proof details in this paper are written to directly rule out relativized succinct arguments with computational soundness (which for simplicity we just refer to as "soundness").

Overall, this extension significantly strengthens the impossibility results in this paper, putting them in sharp contrast with the aforementioned possibility results in other oracle models [CT10; CCS22; CCGOS23].

2.8 Limitations in other oracle models?

The results in this paper rule out relativized succinct arguments in the ROM, even with only computational soundness. The landscape of relativized succinct arguments in other oracle models of interest is not clear.

For example, consider the low-degree random oracle model (LDROM), which is the distribution of random oracles extended to some given low degree. [CCS22] constructs relativized SNARGs in the LDROM with computational soundness (the construction relies on a collision-resistant hash function). Hence one cannot prove impossibility results for the LDROM that are analogous to those we prove for the ROM; at best, one could hope to rule out relativized succinct arguments in the LDROM with *statistical* (rather than computational) soundness.

However, the techniques we used for impossibility results in the ROM do not seem useful for the LDROM. As sketched in Section 2.3, the "hardness" of a random oracle comes from the fact that two oracles with small distance may determine differently the membership of a given instance in the relativized language. In contrast, in the LDROM, oracles are low-degree polynomials, which are far from one another.

This notwithstanding, [CL20] proves that probabilistic proofs in the LDROM do not exist, via a somewhat different approach: they consider a subset of low-degree random oracles that admits "hardness". In particular, they identify a subset of low-degree polynomials that are hard to distinguish from the all-zero polynomial for any query-bounded algorithm and establish some weaker separation result (somewhere separation instead of almost-everywhere separation) in the LDROM for NTIME and PCP. We do not see how their approach could be made to work for SNARGs because the SNARG might fail over the hard subset of oracles but still behave nicely on average (the hard set has exponentially small measure).

Overall, the case of the LDROM remains not clear, and highlights that the study of relativized probabilistic proofs and of relativized succinct arguments can lead to completely different answers, from which we learn that the two research questions are qualitatively different.

A similar open question exists for the arithmetized random oracle model (AROM) in [CCGOS23]: they construct relativized SNARGs in the AROM with computational soundness, but it is unknown if relativized SNARGs in the AROM with statistical soundness exist or not.

3 Preliminaries

3.1 Languages and relations

Definition 3.1 (Language)**.** *A language L is a set of instances $\mathbb{x}$.*

Definition 3.2 (Relation)**.** *A relation R is a set of tuples of instance and witness $(\mathbb{x}, \mathbb{w})$.*

Definition 3.3 (Oracle)**.** *An oracle is a collection of distributions over functions, with one distribution per output length. An* **oracle** *is a collection $\mathcal{U} = \{\mathcal{U}_\ell\}_{\ell \in \mathbb{N}}$ where each $\mathcal{U}_\ell$ is a distribution over functions $u \colon \{0,1\}^* \to \{0,1\}^\ell$.*

For each $\ell \in \mathbb{N}$, we write $u \leftarrow \mathcal{U}_\ell$ to denote that u is a sample of the distribution $\mathcal{U}_\ell$. We write $u \in \mathcal{U}_\ell$ to denote that u is in the support of the distribution $\mathcal{U}_\ell$.

Each $\mathcal{U}_\ell$ induces a corresponding probability measure $\mu_{\mathcal{U}_\ell}(X)$ for a set X of functions from $\{0,1\}^* \to \{0,1\}^\ell$, which is the probability that a sample u from $\mathcal{U}_\ell$ belongs to X.

Definition 3.4 (relativized language)**.** *Let $\mathcal{U}$ be an oracle. An relativized language $L_{\mathcal{U}}$ is a collection of languages indexed by output length $\ell \in \mathbb{N}$ and functions $u \in \mathcal{U}_\ell$, namely,*

$$L_{\mathcal{U}} := \{L_u\}_{\ell \in \mathbb{N}, u \in \mathcal{U}_\ell} \ .$$

Definition 3.5 (relativized relation)**.** *Let $\mathcal{U}$ be an oracle. An relativized relation $R_{\mathcal{U}}$ is a collection of relations indexed by output length $\ell \in \mathbb{N}$ and functions $u \in \mathcal{U}_\ell$, namely,*

$$R_{\mathcal{U}} := \{R_u\}_{\ell \in \mathbb{N}, u \in \mathcal{U}_\ell} \ .$$

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

$$L(R_{\mathcal{U}}) := \{L(R_u)\}_{\ell \in \mathbb{N}, u \in \mathcal{U}_\ell} \ ,$$

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

Definition 3.6. *A* **random oracle** *$\mathcal{O}$ is defined as follows:*

$$\mathcal{O} := \{\mathcal{O}_\ell\}_{\ell \in \mathbb{N}} \ ,$$

where each $\mathcal{O}_\lambda$ is the uniform distribution over functions $f \colon \{0,1\}^* \to \{0,1\}^\ell$.

3.2 Oracle Turing machines

We consider Turing machines that can query oracles. An oracle machine is a machine that has *black-box* access to a function $f : \{0,1\}^* \to \{0,1\}^*$ that the machine can query. Each query costs the machine a single computational step.

Definition 3.7. *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 $x \in \{0,1\}^*$ and oracle $f : \{0,1\}^* \to \{0,1\}^*$, $M^f(\mathbb{x})$ works as follows:*

- *The input x is written in the input tape.*
- *The execution of M proceeds as a normal Turing machine computation except when M intends to make a query to the oracle, then it enters the* QUERY *state.*

- *In the* QUERY *state, M writes its query $q \in \{0,1\}^*$ in the query tape.*
- *In the following step, M enters the* ANSWER *state and the content in the answer tape becomes $f(q) \in \{0,1\}^*$.*

We consider 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.

3.3 Complexity classes with oracles

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 $(\mathbb{x}, \mathbb{w}) \in R$.

Definition 3.8 (NTIME)**.** *Let t be a time-constructible function. The complexity class $\mathsf{NTIME}[t]$ is the set of relations R such that there exists a ND-decider that runs in time $O(t)$.*

Definition 3.9 (Relativized NTIME)**.** *Let $\mathcal{U}$ be an oracle. Let t be a time-constructible function. The complexity class relativized $\mathsf{NTIME}^{\mathcal{U}}[t]$ is the set of relativized relations $R_{\mathcal{U}} = \{R_u\}_{\ell \in \mathbb{N}, u \in \mathcal{U}_\ell}$ such that there exists an oracle nondeterministic machine M that runs in time $O(t)$ and, for every $\ell \in \mathbb{N}$,*

$$\Pr\left[M^u \text{ is an ND-decider for } R_u \mid u \leftarrow \mathcal{U}_\ell\right] = 1 \ .$$

Deterministic time. A deterministic machine M is a decider for a language L if for every $\mathbb{x}$ it holds that $M(\mathbb{x}) = 1$ if and only if $\mathbb{x} \in L$.

Definition 3.10 (DTIME)**.** *Let t be a time-constructible function. The complexity class $\mathsf{DTIME}[t]$ is the set of languages L such that there exists a ND-decider that runs in time $O(t)$.*

Definition 3.11 (Relativized DTIME)**.** *Let $\mathcal{U}$ be an oracle. Let t be a time-constructible function. The complexity class relativized $\mathsf{DTIME}^{\mathcal{U}}[t]$ is the set of relativized relations $L_{\mathcal{U}} = \{L_u\}_{\ell \in \mathbb{N}, u \in \mathcal{U}_\ell}$ such that there exists an oracle deterministic machine M that runs in time $O(t)$ and, for every $\ell \in \mathbb{N}$,*

$$\Pr\left[M^u \text{ is a decider for } L_u \mid u \leftarrow \mathcal{U}_\ell\right] = 1 \ .$$

3.4 Relativized non-interactive arguments in the ROM

A **relativized non-interactive argument** in the ROM (random oracle model) for an relativized relation $R_{\mathcal{O}}$ is a tuple of algorithms $\mathsf{NARG} = (\mathcal{P}, \mathcal{V})$ that works as follows: For every $\lambda \in \mathbb{N}$ and $f \in \mathcal{O}_\lambda$,

- $\mathcal{P}^f(1^\lambda, \mathbb{x}, \mathbb{w}) \to \pi$: Given oracle access to $f \colon \{0,1\}^* \to \{0,1\}^\lambda$, on input the security parameter $\lambda \in \mathbb{N}$, an instance $\mathbb{x}$ and a witness $\mathbb{w}$, the prover $\mathcal{P}$ computes an argument string π that attests to the claim that $(\mathbb{x}, \mathbb{w}) \in R_f$.
- $\mathcal{V}^f(1^\lambda, \mathbb{x}, \pi) \to b$: Given oracle access to $f \colon \{0,1\}^* \to \{0,1\}^\lambda$, on input the security parameter $\lambda \in \mathbb{N}$, an instance $\mathbb{x}$ and a corresponding argument string π, the verifier $\mathcal{V}$ outputs a decision a bit b.

Definition 3.12 (Completeness)**.** *For every security parameter $\lambda \in \mathbb{N}$, instance size bound $n \in \mathbb{N}$, and adversary $\mathcal{A}$,*

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

Definition 3.13 (Soundness)**.** *For every security parameter $\lambda \in \mathbb{N}$, instance size bound $n \in \mathbb{N}$, 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 $\widetilde{\mathcal{P}}$,*

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

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

- the *argument size* $\mathsf{as}(\lambda, n)$ is the maximum number of bits in the argument string π;
- 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 3.14 (Relativized NARG in the ROM)**.** *Let $\mathcal{O}$ be the random oracle. The complexity class*

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

is the set of relativized relations $R_{\mathcal{O}} = \{R_f\}_{\lambda \in \mathbb{N}, f \in \mathcal{O}_\lambda}$ that admits a relativized non-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 argument size is* as*;*
- *the honest prover query complexity is* pq*;*
- *the honest prover running time is* pt.

4 Separation of DTIME and NARG

Theorem 4.1. *Let $\mathcal{O}$ be the random oracle. For every time-constructible function $t\colon \mathbb{N} \to \mathbb{N}$, argument verifier query bound $\mathsf{vq}\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}$, argument verifier time bound $\mathsf{vt}\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}$, argument size bound $\mathsf{as}\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}$, argument honest prover query bound $\mathsf{pq}\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}$, argument honest prover time bound $\mathsf{pt}\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}$, argument completeness $\alpha\colon \mathbb{N}\times\mathbb{N} \to [0,1]$ and argument soundness $\beta\colon \mathbb{N}\times\mathbb{N}\times\mathbb{N}\times\mathbb{N} \to [0,1]$,*

$$\mathsf{DTIME}^{\mathcal{O}}\left[t(n)\right] \not\subseteq \mathsf{NARG}^{\mathcal{O}}\left[\begin{array}{lccl} \text{completeness error} & \alpha & = & \alpha(\lambda,n) \\ \text{soundness error} & \beta & = & \beta(\lambda,n,\mathsf{q}_{\widetilde{\mathcal{P}}},\mathsf{t}_{\widetilde{\mathcal{P}}}) \\ \text{verifier query bound} & \mathsf{vq} & = & \mathsf{vq}(\lambda,n) \\ \text{verifier running time} & \mathsf{vt} & = & \mathsf{vt}(\lambda,n) \\ \text{argument size} & \mathsf{as} & = & \mathsf{as}(\lambda,n) \\ \text{honest prover query bound} & \mathsf{pq} & = & \mathsf{pq}(\lambda,n) \\ \text{honest prover time bound} & \mathsf{pt} & = & \mathsf{pt}(\lambda,n) \end{array}\right],$$

where there exists $\lambda \in \mathbb{N}$, $n \in \mathbb{N}$ and some function $p\colon \mathbb{N} \to (0,1]$ such that the following holds:

- $0 \le \alpha(\lambda,n) \le 1$,
- $0 \le \beta(\lambda,n,t(n)+\mathsf{pq},t(n)+\mathsf{pt}) < (1-\alpha(\lambda,n)) \cdot \frac{t(n)-\frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)} \cdot (1-p(n))$, *and*
- $(1-\alpha(\lambda,n)) \cdot \frac{t(n)-\frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)} \cdot (1-p(n)) \le 1$.

The corollary below follows from setting $p(n) < \frac{1}{2}$ for all $n \in \mathbb{N}$.

Corollary 4.2. *Let $\mathcal{O}$ be the random oracle. For every time-constructible function $t\colon \mathbb{N}\times\mathbb{N}$, $\lambda \in \mathbb{N}$ and $n \in \mathbb{N}$,*

$$\mathsf{DTIME}^{\mathcal{O}}\left[t(n)\right] \not\subseteq \mathsf{NARG}^{\mathcal{O}}\left[\begin{array}{lccl} \text{completeness error} & \alpha & = & 1/3 \\ \text{soundness error} & \beta & = & \begin{cases} 1/3 & \mathsf{q}_{\widetilde{\mathcal{P}}} = \mathsf{poly}(\lambda,t(n)), \mathsf{t}_{\widetilde{\mathcal{P}}} = \mathsf{poly}(\lambda,t(n)) \\ 1 & \textit{otherwise} \end{cases} \\ \text{verifier query bound} & \mathsf{vq} & = & o(t(n)) \\ \text{verifier running time} & \mathsf{vt} & & \\ \text{argument size} & \mathsf{as} & & \\ \text{honest prover query bound} & \mathsf{pq} & = & \mathsf{poly}(\lambda,t(n)) \\ \text{honest prover time bound} & \mathsf{pq} & = & \mathsf{poly}(\lambda,t(n)) \end{array}\right].$$

4.1 Separation for every security parameter

Note that we discuss relations in DTIME in this section, therefore, we only need to focus on the corresponding languages since the witnesses are always empty. To prove Theorem 4.1, it suffices to find an relativized language $L_{\mathcal{O}_\lambda}$ that can be decided by an oracle deterministic Turing machine in $O(t(n))$ time but does not have a relativized argument system.

Definition 4.3. *Let $u_{k,i}$ denote the $\lceil \log k \rceil$-bit string that represents $i \in [k]$. For every time-constructible function $t\colon \mathbb{N}\times\mathbb{N}$ with $t(n) \ge n$ for every $n \in \mathbb{N}$, $n \in \mathbb{N}$ and $(x,y) \in \{0,1\}^{n/2} \times \{0,1\}^{n/2}$, we define*

$$t^*(n) := \frac{t(n)}{n/2} \ ,$$

and for every $\lambda \in \mathbb{N}$ and $f \in \mathcal{O}_\lambda$ and $i \in [n/2]$,

$$F_{f,n}(x)_i \coloneqq \bigoplus_{j \in \{(i-1)\cdot t^*(n)+1,(i-1)\cdot t^*(n)+2,\cdots,i\cdot t^*(n)\}} f(x \parallel u_{t(n),j})_1 \ .$$

We define $L_\mathcal{O} = \{L_f\}_{\lambda\in\mathbb{N},f\in\mathcal{O}_\lambda}$ as follows:

$$L_f \coloneqq \left\{(x,y) \in \{0,1\}^{n/2} \times \{0,1\}^{n/2} \;\middle|\; F_{f,n}(x) = y\right\} \ .$$

The following two lemmas directly imply Theorem 4.1.

Lemma 4.4. *Let $\mathcal{O}$ be the random oracle. Let $L_\mathcal{O}$ be defined as in Definition 4.3. For every time-constructible function $t\colon \mathbb{N} \times \mathbb{N}$:*

$$L_\mathcal{O} \in \mathsf{DTIME}[t] \ .$$

Lemma 4.5. *Let $\mathcal{O}$ be the random oracle. Let $L_\mathcal{O}$ be defined as in Definition 5.3. Fix time-constructible function $t\colon \mathbb{N} \to \mathbb{N}$, argument verifier query bound $\mathsf{vq}\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}$, argument verifier time bound $\mathsf{vt}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument size bound $\mathsf{as}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument honest prover query bound $\mathsf{pq}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument honest prover time bound $\mathsf{pt}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument completeness $\alpha\colon \mathbb{N}\times\mathbb{N}\to[0,1]$ and argument soundness $\beta\colon \mathbb{N}\times\mathbb{N}\times\mathbb{N}\times\mathbb{N}\to[0,1]$.*

There is no relativized argument $(\mathcal{P},\mathcal{V})$ for the relativized language $L_{\mathcal{O}_\lambda}$ with completeness error α, soundness error β, argument verifier running time vt, argument size as, argument honest prover query bound pq, and argument honest prover time bound pt if there exists $\lambda \in \mathbb{N}$, $n \in \mathbb{N}$ and $p\colon \mathbb{N}\to\{0,1\}$ such that the following holds:

- $0 \le \alpha(\lambda,n) \le 1$,
- $0 \le \beta(\lambda,n,t(n)+\mathsf{pq},t(n)+\mathsf{pt}) < (1-\alpha(\lambda,n)) \cdot \frac{t(n)-\frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)} \cdot (1-p(n))$, *and*
- $(1-\alpha(\lambda,n)) \cdot \frac{t(n)-\frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)} \cdot (1-p(n)) \le 1$.

We prove Lemma 4.4 in Section 4.2 and Lemma 4.5 in Section 4.3.

4.2 Proof of Lemma 4.4

To show that language $L_\mathcal{O}$ is in $\mathsf{DTIME}[(t(n)]$, we construct a deterministic Turing machine M that decides $L_\mathcal{O}$ within $O(t(n))$ time.

$M^f((x,y))$:
1. For every $i \in [n/2]$: Query the oracle f at $\{x \parallel u_{t(n),j}\}_{j\in\{(i-1)\cdot t^*(n)+1,\cdots,i\cdot t^*(n)\}}$ to compute $F_{f,n}(x)_i$.
2. Check if $y = F_{f,n}(x)$.

It is obvious that $M^f((x,y)) = 1$ if and only if $(x,y) \in L_f$ for every $\lambda \in \mathbb{N}$ and $f \in \mathcal{O}_\lambda$.

Now we argue that M^f decides L_f within time $O(t(n))$. On input $\mathbb{x} \coloneqq (x,y) \in \{0,1\}^{n/2}\times\{0,1\}^{n/2}$ and $f \in \mathcal{O}_\lambda$, M^f writes x on the query tape, which takes time $O(n)$. M then queries the oracle f with $x \parallel u_{t(n),j}$ for all each $i \in [n/2]$ and $j \in \{(i-1)\cdot t^*(n)+1,\cdots,i\cdot t^*(n)\}$, which takes amortized time $O(t(n))$ in total. Computing $F_{f,n}(x)$ takes time $O(t(n))$. Finally, comparing y and $F_{f,n}(x)$ takes time $O(n)$.

4.3 Proof of Lemma 4.5

Assume for the sake of contradiction that $L_{\mathcal{O}}$ has such a relativized argument system $(\mathcal{P}, \mathcal{V})$ and there exists $\lambda \in \mathbb{N}$, $n \in \mathbb{N}$ and $p \colon \mathbb{N} \to [0,1]$ that satisfy the conditions in Lemma 4.5.

For every $i \in [n/2]$ and $j \in [t^*(n)]$, let $q(i,j)$ be defined as follows:

$$q(i,j) \coloneqq 0^{n/2} \parallel u_{t(n),(i-1)\cdot t^*(n)+j} \enspace .$$

Consider the set of queries Q:

$$Q \coloneqq \{q(i,j)\}_{i\in[n/2],j\in[t^*(n)]} .$$

Note that $|Q| = t(n)$.

For every $f \in \mathcal{O}_\lambda$ and $y \in \{0,1\}^{n/2}$, define the set of queries with "low" probability:

$$Q^\star(f,y) \coloneqq \left\{ q \in Q : \Pr\left[\mathcal{V}^f(1^\lambda, (0^{n/2}, y), \pi, \rho) \text{ queries } f \text{ at } q \;\middle|\; \begin{array}{l} \pi \leftarrow \mathcal{P}^f(1^\lambda, (0^{n/2}, y)) \\ \rho \leftarrow \{0,1\}^{\mathsf{vr}} \end{array} \right] < p(n) \right\} \enspace .$$

Consider the set of queries with "high" probability:

$$Q_c(f,y) \coloneqq \left\{ q \in Q : \Pr\left[\mathcal{V}^f(1^\lambda, (0^{n/2}, y), \pi, \rho) \text{ queries } f \text{ at } q \;\middle|\; \begin{array}{l} \pi \leftarrow \mathcal{P}^f(1^\lambda, (0^{n/2}, y)) \\ \rho \leftarrow \{0,1\}^{\mathsf{vr}} \end{array} \right] \geq p(n) \right\} \enspace .$$

Since $\mathcal{V}$ makes at most $\mathsf{vq}(\lambda, n)$ queries to f for every $f \in \mathcal{O}_\lambda$,

$$\sum_{q\in Q} \Pr\left[\mathcal{V}^f(1^\lambda, (0^{n/2}, y), \pi, \rho) \text{ queries } f \text{ at } q \;\middle|\; \begin{array}{l} \pi \leftarrow \mathcal{P}^f(1^\lambda, (0^{n/2}, y)) \\ \rho \leftarrow \{0,1\}^{\mathsf{vr}} \end{array} \right] \leq \mathsf{vq}(\lambda, n) \enspace .$$

Hence,

$$|Q_c(f,y)| \leq \frac{\mathsf{vq}(\lambda, n)}{p(n)} \enspace .$$

We can conclude that

$$|Q^\star(f,y)| = t(n) - |Q_c(f,y)| \geq t(n) - \frac{\mathsf{vq}(\lambda, n)}{p(n)} \enspace . \tag{3}$$

On the other hand, by definition, for every $i \in [n/2]$ and $j \in [t^*(n)]$ such that $q = q(i,j) \in Q^\star(f,y)$, it holds that

$$\Pr\left[\mathcal{V}^f(1^\lambda, (0^{n/2}, y), \pi, \rho) \text{ queries } f \text{ at } q \;\middle|\; \begin{array}{l} \pi \leftarrow \mathcal{P}^f(1^\lambda, (0^{n/2}, y)) \\ \rho \leftarrow \{0,1\}^{\mathsf{vr}} \end{array} \right] < p(n) \enspace ,$$

which implies that

$$\Pr\left[\mathcal{V}^f(1^\lambda, (0^{n/2}, y), \pi, \rho) \text{ doesn't query } f \text{ at } q \;\middle|\; \begin{array}{l} \pi \leftarrow \mathcal{P}^f(1^\lambda, (0^{n/2}, y)) \\ \rho \leftarrow \{0,1\}^{\mathsf{vr}} \end{array} \right] \geq 1 - p(n) \enspace . \tag{4}$$

For every $f \in \mathcal{O}_\lambda$, $i \in [n/2]$ and $j \in [t^*(n)]$, we define $\mathsf{Flip}(f,i,j)$ as follows:

$\mathsf{Flip}(f,i,j)$:
1. Set $f' \coloneqq f$.
2. Set $f'(q(i,j))_1 = 1 - f(q(i,j))_1$.

3. Output f'.

We define our malicious prover $\widetilde{\mathcal{P}}_{n,i,j}$:

$\widetilde{\mathcal{P}}^f_{n,i,j}$:
1. Sample randomness for honest prover $\mathcal{P}$: $\zeta \leftarrow \{0,1\}^{\mathsf{pr}}$.
2. Let $x \coloneqq 0^{n/2}$.
3. Query f at $x \parallel u_{t(n),m}$ for all $m \in [t(n)]$ to compute $y \coloneqq F_{f,n}(x)$.
4. Set $y_i \coloneqq \left(\bigoplus_{j' \in \{(i-1)\cdot t^*(n)+1,\cdots,i\cdot t^*(n)\}\setminus\{(i-1)\cdot t^*(n)+j\}} f(x \parallel u_{t(n),j'})_1\right) \oplus (1 - f(x \| u_{t(n),(i-1)\cdot t^*(n)+j})_1)$.
5. Simulate $\mathcal{P}(1^\lambda, (x,y), \zeta)$.
6. If $\mathcal{P}$ makes a query to $x \parallel u_{t(n),(i-1)\cdot t^*(n)+j}$:
   (a) Let $\mathsf{ans} \coloneqq f(x \parallel u_{t(n),(i-1)\cdot t^*(n)+j})$.
   (b) Let $\mathsf{ans}' \coloneqq \mathsf{ans}$.
   (c) Set $\mathsf{ans}'_1 \coloneqq 1 - \mathsf{ans}_1$.
   (d) Answer $\mathcal{P}$'s query by ans'.
7. Let π' be the proof outputted by $\mathcal{P}(1^\lambda, (x,y), \zeta)$ with altered oracle query's answer.
8. Output $((x,y),\pi')$.

Note that $\widetilde{\mathcal{P}}_{n,i,j}$ makes at most $t(n) + \mathsf{pq}$ queries to f and runs in time at most $t(n) + \mathsf{pt}$.

Consider the adversary $\mathcal{A}_n$:

$\mathcal{A}^f_n$:
1. Let $x \coloneqq 0^{n/2}$.
2. Set $y \coloneqq F_{f,n}(x)$.
3. Output (x,y).

For the derivation below, we omit the sampling of the verifier's randomness ρ and prover's randomness ζ in the experiment, they are always sampled as follows:

$$\left[\begin{array}{l} \zeta \leftarrow \{0,1\}^{\mathsf{pr}} \\ \rho \leftarrow \{0,1\}^{\mathsf{vr}} \end{array}\right] .$$

Then, we can deduce the following:

$$\Pr\left[\begin{array}{l|l} (0^{n/2}, y') \notin L_{f'} & f \leftarrow \mathcal{O}_\lambda \\ \wedge\, \mathcal{V}^{f'}(1^\lambda, (0^{n/2}, y'), \widetilde{\pi}, \rho) = 1 & (i,j) \leftarrow [n/2] \times [t^*(n)] \\ & f' \coloneqq \mathsf{Flip}(f,i,j) \\ & (0^{n/2}, y', \widetilde{\pi}) \coloneqq \widetilde{\mathcal{P}}^{f'}_{n,i,j}(\zeta) \end{array}\right]$$

$$=_{[1]} \Pr\left[\begin{array}{l|l} & f \leftarrow \mathcal{O}_\lambda \\ \mathcal{V}^{f'}(1^\lambda, (0^{n/2}, y'), \widetilde{\pi}, \rho) = 1 & (i,j) \leftarrow [n/2] \times [t^*(n)] \\ & f' \coloneqq \mathsf{Flip}(f,i,j) \\ & (0^{n/2}, y', \widetilde{\pi}) \coloneqq \widetilde{\mathcal{P}}^{f'}_{n,i,j}(\zeta) \end{array}\right]$$

$$\geq_{[2]} \Pr\left[\begin{array}{l|l} & f \leftarrow \mathcal{O}_\lambda \\ \mathcal{V}^{f'}(1^\lambda, (0^{n/2}, y'), \widetilde{\pi}, \rho) = 1 & (i,j) \leftarrow [n/2] \times [t^*(n)] \\ \text{conditioned on} & f' \coloneqq \mathsf{Flip}(f,i,j) \\ \mathcal{V}^{f}(1^\lambda, (0^{n/2}, y), \pi, \rho) = 1 & (0^{n/2}, y', \widetilde{\pi}) \coloneqq \widetilde{\mathcal{P}}^{f'}_{n,i,j}(\zeta) \\ \wedge\, q(i,j) \in Q^\star(f,y) & (0^{n/2}, y) \coloneqq \mathcal{A}^f_n \\ & \pi \coloneqq \mathcal{P}^f(1^\lambda, (0^{n/2}, y), \zeta) \end{array}\right]$$

$$\cdot \Pr\left[\begin{array}{l|l} \mathcal{V}^f(1^\lambda,(0^{n/2},y),\pi,\rho)=1 & f\leftarrow\mathcal{O}_\lambda \\ \wedge\, q(i,j)\in Q^\star(f,y) & (i,j)\leftarrow[n/2]\times[t^*(n)] \\ & (0^{n/2},y) \coloneqq \mathcal{A}_n^f \\ & \pi \coloneqq \mathcal{P}^f(1^\lambda,(0^{n/2},y),\zeta) \end{array}\right]$$

$$\geq_{[3]} (1-p(n))\cdot \Pr\left[\begin{array}{l|l} \mathcal{V}^f(1^\lambda,(0^{n/2},y),\pi,\rho)=1 & f\leftarrow\mathcal{O}_\lambda \\ \wedge\, q(i,j)\in Q^\star(f,y) & (i,j)\leftarrow[n/2]\times[t^*(n)] \\ & (0^{n/2},y) \coloneqq \mathcal{A}_n^f \\ & \pi \coloneqq \mathcal{P}^f(1^\lambda,(0^{n/2},y),\zeta) \end{array}\right],$$

where

- Equality [1] holds because for every $f\in\mathcal{O}_\lambda$, $i\in[n]$ and $j\in[t^*(n)]$, $\widetilde{\mathcal{P}}_n^{\mathsf{Flip}(f,i,j)}$ outputs instance $(x',y')=(0^n,F_{f,n}(0^n))\notin L_{\mathsf{Flip}(f,i,j)}$;
- Inequality [2] holds by definition of conditional probability;
- Inequality [3] holds by Eq. (4) and that for every $f\in\mathcal{O}_\lambda$, $(i,j)\in[n/2]\times[t^*(n)]$, $(x,y)\in\{0,1\}^{n/2}\times\{0,1\}^{n/2}$ and $\zeta\in\{0,1\}^{\mathsf{pr}}$, $\mathcal{P}^f(1^\lambda,(x,y),\zeta)$ and $\widetilde{\mathcal{P}}_{n,i,j}^{\mathsf{Flip}(f,i,j)}(\zeta)$ output the same proof. Moreover, $\widetilde{\mathcal{P}}_{n,i,j}^{\mathsf{Flip}(f,i,j)}(\zeta)$ and $\mathcal{A}_n^f$ output the same instance.

Moreover,

$$\Pr\left[\begin{array}{l|l} \mathcal{V}^f(1^\lambda,(0^{n/2},y),\pi,\rho)=1 & f\leftarrow\mathcal{O}_\lambda \\ \wedge\, q(i,j)\in Q^\star(f,y) & (i,j)\leftarrow[n/2]\times[t^*(n)] \\ & (0^{n/2},y) \coloneqq \mathcal{A}_n^f \\ & \pi \coloneqq \mathcal{P}^f(1^\lambda,(0^{n/2},y),\zeta) \end{array}\right]$$

$$= \sum_{\substack{f^\star\in\mathcal{O}_\lambda \\ i^\star\in[n/2] \\ j^\star\in[t^*(n)] \\ y\coloneqq F_{f^\star,n}(0^{n/2}) \\ q(i^\star,j^\star)\in Q^\star(f^\star,y)}} \left(\begin{array}{l} \Pr\left[\begin{array}{l|l} f=f^\star & f\leftarrow\mathcal{O}_\lambda \\ (i,j)=(i^\star,j^\star) & (i,j)\leftarrow[n/2]\times[t^*(n)] \end{array}\right] \\ \cdot\Pr\left[\mathcal{V}^{f^\star}(1^\lambda,(0^{n/2},y),\pi,\rho)=1 \,\middle|\, \begin{array}{l} (0^{n/2},y)\coloneqq\mathcal{A}_n^{f^\star} \\ \pi\coloneqq\mathcal{P}^{f^\star}(1^\lambda,(0^{n/2},y),\zeta) \end{array}\right] \end{array}\right)$$

$$=_{[1]} \frac{1}{t(n)}\cdot \sum_{\substack{f^\star\in\mathcal{O}_\lambda \\ i^\star\in[n/2] \\ j^\star\in[t^*(n)] \\ y\coloneqq F_{f^\star,n}(0^{n/2}) \\ q(i^\star,j^\star)\in Q^\star(f^\star,y)}} \Pr\left[f=f^\star \mid f\leftarrow\mathcal{O}_\lambda\right]\cdot\Pr\left[\mathcal{V}^{f^\star}(1^\lambda,(0^{n/2},y),\pi,\rho)=1 \,\middle|\, \begin{array}{l} (0^{n/2},y)\coloneqq\mathcal{A}_n^{f^\star} \\ \pi\coloneqq\mathcal{P}^{f^\star}(1^\lambda,(0^{n/2},y),\zeta) \end{array}\right]$$

$$=_{[2]} \frac{1}{t(n)}\cdot \sum_{\substack{f^\star\in\mathcal{O}_\lambda \\ i^\star\in[n/2] \\ j^\star\in[t^*(n)] \\ y\coloneqq F_{f^\star,n}(0^{n/2}) \\ q(i^\star,j^\star)\in Q^\star(f^\star,y)}} \Pr\left[f=f^\star \mid f\leftarrow\mathcal{O}_\lambda\right]\cdot\Pr\left[\begin{array}{l|l} (0^{n/2},y)\notin L_{f^\star} & (0^{n/2},y)\coloneqq\mathcal{A}_n^{f^\star} \\ \vee\,\mathcal{V}^{f^\star}(1^\lambda,(0^{n/2},y),\pi,\rho)=1 & \pi\coloneqq\mathcal{P}^{f^\star}(1^\lambda,(0^{n/2},y),\zeta) \end{array}\right]$$

$$\geq_{[3]} \frac{t(n)-\frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)}\cdot \sum_{f^\star\in\mathcal{O}_\lambda} \Pr\left[f=f^\star \mid f\leftarrow\mathcal{O}_\lambda\right]\cdot\Pr\left[\begin{array}{l|l} (0^{n/2},y)\notin L_{f^\star} & (0^{n/2},y)\coloneqq\mathcal{A}_n^{f^\star} \\ \vee\,\mathcal{V}^{f^\star}(1^\lambda,(0^{n/2},y),\pi,\rho)=1 & \pi\coloneqq\mathcal{P}^{f^\star}(1^\lambda,(0^{n/2},y),\zeta) \end{array}\right]$$

$$
= \frac{t(n) - \frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)} \cdot \Pr\left[\begin{array}{l} (0^{n/2}, y) \notin L_f \\ \vee \mathcal{V}^f(1^\lambda, (0^{n/2}, y), \pi, \rho) = 1 \end{array} \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^{n/2}, y) := \mathcal{A}_n^f \\ \pi := \mathcal{P}^f((0^{n/2}, y), \zeta) \end{array} \right]
$$

$$
\geq_{[4]} \frac{t(n) - \frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)} \cdot (1 - \alpha(\lambda, n)) \ ,
$$

where

- Equality [1] holds because for every $(i^\star, j^\star) \in [n/2] \times [t^*(n)]$, $\Pr\left[(i,j) = (i^\star, j^\star) \mid (i,j) \leftarrow [n/2] \times [t^*(n)]\right] = \frac{1}{t(n)}$;
- Equality [2] holds because for every $n \in \mathbb{N}$ and $f \in \mathcal{O}_\lambda$, $\mathcal{A}_n^f$ outputs $(0^n, y) = (0^{n/2}, F_{f,n}(0^{n/2})) \in L_f$. Hence, $(0^n, y) \notin L_f$ always happens with probability 0;
- Inequality [3] follows from Eq. (3);
- Inequality [4] follows from Definition 3.12.

On the other hand,

$$
\Pr\left[\begin{array}{l} (0^{n/2}, y') \notin L_{f'} \\ \wedge \mathcal{V}^{f'}(1^\lambda, (0^{n/2}, y'), \widetilde{\pi}, \rho) = 1 \end{array} \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (i,j) \leftarrow [n/2] \times [t^*(n)] \\ f' := \mathsf{Flip}(f, i, j) \\ (0^{n/2}, y', \widetilde{\pi}) := \widetilde{\mathcal{P}}_{n,i,j}^{f'}(\zeta) \end{array} \right]
$$

$$
= \sum_{\substack{f^\star \in \mathcal{O}_\lambda \\ (i^\star, j^\star) \in [n/2] \times [t^*(n)] \\ f' := \mathsf{Flip}(f^\star, i^\star, j^\star)}} \left(\begin{array}{l} \Pr\left[\begin{array}{l} f = f^\star \\ \wedge (i,j) = (i^\star, j^\star) \end{array} \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (i,j) \leftarrow [n/2] \times [t^*(n)] \end{array} \right] \\ \cdot \Pr\left[\begin{array}{l} (0^{n/2}, y') \notin L_{f'} \\ \wedge \mathcal{V}^{f'}(1^\lambda, (0^{n/2}, y'), \widetilde{\pi}, \rho) = 1 \end{array} \;\middle|\; (0^{n/2}, y', \widetilde{\pi}) := \widetilde{\mathcal{P}}_{n,i^\star,j^\star}^{f'}(\zeta) \right] \end{array} \right)
$$

$$
=_{[1]} \sum_{\substack{(i^\star, j^\star) \in [n/2] \times [t^*(n)] \\ f^\star \in \mathcal{O}_\lambda}} \left(\begin{array}{l} \Pr\left[\begin{array}{l} f = f^\star \\ \wedge (i,j) = (i^\star, j^\star) \end{array} \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (i,j) \leftarrow [n/2] \times [t^*(n)] \end{array} \right] \\ \cdot \Pr\left[\begin{array}{l} (0^{n/2}, y') \notin L_{f^\star} \\ \wedge \mathcal{V}^{f^\star}(1^\lambda, (0^{n/2}, y'), \widetilde{\pi}, \rho) = 1 \end{array} \;\middle|\; (0^{n/2}, y', \widetilde{\pi}) := \widetilde{\mathcal{P}}_{n,i^\star,j^\star}^{f^\star}(\zeta) \right] \end{array} \right)
$$

$$
= \frac{1}{t(n)} \sum_{\substack{(i^\star, j^\star) \in [n/2] \times [t^*(n)] \\ f^\star \in \mathcal{O}_\lambda}} \left(\begin{array}{l} \Pr\left[f = f^\star \mid f \leftarrow \mathcal{O}_\lambda\right] \\ \cdot \Pr\left[\begin{array}{l} (0^{n/2}, y') \notin L_{f^\star} \\ \wedge \mathcal{V}^{f^\star}(1^\lambda, (0^{n/2}, y'), \widetilde{\pi}, \rho) = 1 \end{array} \;\middle|\; (0^{n/2}, y', \widetilde{\pi}) := \widetilde{\mathcal{P}}_{n,i^\star,j^\star}^{f^\star}(\zeta) \right] \end{array} \right)
$$

$$
\leq \frac{1}{t(n)} \cdot t(n) \max_{(i^\star, j^\star) \in [n/2] \times [t^*(n)]} \left\{ \sum_{f^\star \in \mathcal{O}_\lambda} \left(\begin{array}{l} \Pr\left[f = f^\star \mid f \leftarrow \mathcal{O}_\lambda\right] \\ \cdot \Pr\left[\begin{array}{l} (0^{n/2}, y') \notin L_{f^\star} \\ \wedge \mathcal{V}^{f^\star}(1^\lambda, (0^{n/2}, y'), \widetilde{\pi}, \rho) = 1 \end{array} \;\middle|\; (0^{n/2}, y', \widetilde{\pi}) := \widetilde{\mathcal{P}}_{n,i^\star,j^\star}^{f^\star}(\zeta) \right] \end{array} \right) \right\}
$$

$$
= \max_{(i^\star, j^\star) \in [n/2] \times [t^*(n)]} \left\{ \Pr\left[\begin{array}{l} (0^{n/2}, y') \notin L_f \\ \wedge \mathcal{V}^f(1^\lambda, (0^{n/2}, y'), \widetilde{\pi}, \rho) = 1 \end{array} \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^{n/2}, y', \pi') := \widetilde{\mathcal{P}}_{n,i^\star,j^\star}^f(\zeta) \end{array} \right] \right\}
$$

$$
\leq_{[2]} \beta(\lambda, n, t(n) + \mathsf{pq}, t(n) + \mathsf{pt}) \ ,
$$

where

- Equality [1] holds because for every $f \in \mathcal{O}_\lambda$, $i \in [n]$ and $j \in [t^*(n)]$, $\mathsf{Flip}(\mathsf{Flip}(f,i,j),i,j) = f$. Therefore, for every $(i^\star, j^\star) \in [n/2] \times [t^*(n)]$, summing over $f' := \mathsf{Flip}(f^\star, i^*, j^*)$ for every $f^\star \in \mathcal{O}_\lambda$ is equivalent to summing over $f^\star$ for every $f^\star \in \mathcal{O}_\lambda$.
- Inequality [2] follows from Definition 3.13.

Therefore, we can conclude that

$$(1-\alpha(\lambda,n)) \cdot \frac{t(n) - \frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)} \cdot (1-p(n)) \le \beta(\lambda, n, t(n) + \mathsf{pq}, t(n) + \mathsf{pt}) \ ,$$

a contradiction.

5 Separation of NTIME and NARG

Theorem 5.1. *Let $\mathcal{O}$ be the random oracle. For every time-constructible function $t\colon \mathbb{N}\to\mathbb{N}$, argument verifier query bound $\mathsf{vq}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument verifier time bound $\mathsf{vt}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument size bound $\mathsf{as}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument honest prover query bound $\mathsf{pq}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument honest prover time bound $\mathsf{pt}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument completeness $\alpha\colon \mathbb{N}\times\mathbb{N}\to[0,1]$ and argument soundness $\beta\colon \mathbb{N}\times\mathbb{N}\times\mathbb{N}\times\mathbb{N}\to[0,1]$,*

$$\mathsf{NTIME}^{\mathcal{O}}\left[t(n)\right] \not\subseteq \mathsf{NARG}^{\mathcal{O}}\left[\begin{array}{lccl}\text{completeness error} & \alpha & = & \alpha(\lambda,n)\\ \text{soundness error} & \beta & = & \beta(\lambda,n,\mathsf{q}_{\widetilde{\mathcal{P}}},\mathsf{t}_{\widetilde{\mathcal{P}}})\\ \text{verifier query bound} & \mathsf{vq} & = & \mathsf{vq}(\lambda,n)\\ \text{verifier running time} & \mathsf{vt} & = & \mathsf{vt}(\lambda,n)\\ \text{argument size} & \mathsf{as} & = & \mathsf{as}(\lambda,n)\\ \text{honest prover query bound} & \mathsf{pq} & = & \mathsf{pq}(\lambda,n)\\ \text{honest prover time bound} & \mathsf{pt} & = & \mathsf{pt}(\lambda,n)\end{array}\right],$$

where there exists $\lambda\in\mathbb{N}$, $n\in\mathbb{N}$ and some function $p\colon\mathbb{N}\to(0,1]$ such that the following holds:

- $0 \le \beta(\lambda,n,\mathsf{pq},\mathsf{pt}) < \left(1-\frac{1}{2^{t(n)}}\right)^{2^{t(n)}}$,
- $0 \le \frac{2^{t(n)}-1}{2^{t(n)}}\cdot\alpha(\lambda,n) < (1-t(n)\cdot p(n))\cdot\frac{2^{t(n)}-\frac{2^{O(\mathsf{as}(\lambda,n))}\cdot\mathsf{vq}(\lambda,n)}{p(n)}}{2^{t(n)}}\cdot\left(\left(1-\frac{1}{2^{t(n)}}\right)^{2^{t(n)}}-\beta(\lambda,n,\mathsf{pq},\mathsf{pt})\right)$, *and*
- $(1-t(n)\cdot p(n))\cdot\frac{2^{t(n)}-\frac{2^{O(\mathsf{as}(\lambda,n))}\cdot\mathsf{vq}(\lambda,n)}{p(n)}}{2^{t(n)}}\cdot\left(\left(1-\frac{1}{2^{t(n)}}\right)^{2^{t(n)}}-\beta(\lambda,n,\mathsf{pq},\mathsf{pt})\right)\le 1.$

The corollary below follows by setting $p(n) < \frac{1}{2t(n)}$ for all $n\in\mathbb{N}$ in Theorem 5.1.

Corollary 5.2. *Let $\mathcal{O}$ be the random oracle. For every time-constructible function $t\colon \mathbb{N}\times\mathbb{N}$, $\lambda\in\mathbb{N}$ and $n\in\mathbb{N}$,*

$$\mathsf{NTIME}^{\mathcal{O}}\left[t(n)\right] \not\subseteq \ \mathsf{NARG}^{\mathcal{O}}\left[\begin{array}{lccl}\text{completeness error} & \alpha & = & 1/3\\ \text{soundness error} & \beta & = & \begin{cases}1/3 & \mathsf{q}_{\widetilde{\mathcal{P}}}=\mathsf{poly}(\lambda,t(n)),\mathsf{t}_{\widetilde{\mathcal{P}}}=\mathsf{poly}(\lambda,t(n))\\ 1 & \textit{otherwise}\end{cases}\\ \text{verifier query bound} & \mathsf{vq} & = & \mathsf{poly}(\lambda,t(n))\\ \text{verifier running time} & \mathsf{vt} & & \\ \text{argument size} & \mathsf{as} & = & o(t(n))\\ \text{honest prover query bound} & \mathsf{pq} & = & \mathsf{poly}(\lambda,t(n))\\ \text{honest prover time bound} & \mathsf{pq} & = & \mathsf{poly}(\lambda,t(n))\end{array}\right].$$

5.1 Separation for every security parameter

To prove Theorem 5.1, it is enough to find one relativized relation $R_{\mathcal{O}}$ that can be decided by an oracle nondeterministic Turing machine within $O(t(n))$ time but does not have a relativized argument system.

We first define the relativized relation we investigate.

Definition 5.3. *Let $u_{k,i}$ denote the $\lceil \log k \rceil$-bit string that represents $i \in [k]$. For every time-constructible function $t\colon \mathbb{N} \times \mathbb{N}$ with $t(n) \geq n$ for every $n \in \mathbb{N}$, we define $R_{\mathcal{O}} = \{R_f\}_{\lambda \in \mathbb{N}, f \in \mathcal{O}_\lambda}$ as follows:*

$$R_f := \left\{ (\mathbb{x}, \mathbb{w}) \in \{0,1\}^n \times \{0,1\}^{t(n)} \;\middle|\; \begin{array}{l} \mathbb{x} = 0^n \\ \wedge \forall i \in [t(n)], f(\mathbb{w} \parallel u_{t(n),i})_1 = 0 \end{array} \right\} \ .$$

The following two lemmas directly imply Theorem 5.1.

Lemma 5.4. *Let $\mathcal{O}$ be the random oracle. Let $R_{\mathcal{O}}$ be defined as in Definition 5.3. For every time-constructible function $t\colon \mathbb{N} \times \mathbb{N}$:*

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

Lemma 5.5. *Let $\mathcal{O}$ be the random oracle. Let $R_{\mathcal{O}}$ be defined as in Definition 5.3. Fix time-constructible function $t\colon \mathbb{N} \to \mathbb{N}$, argument verifier query bound $\mathsf{vq}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument verifier time bound $\mathsf{vt}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument size bound $\mathsf{as}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument honest prover query bound $\mathsf{pq}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument honest prover time bound $\mathsf{pt}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument completeness $\alpha\colon \mathbb{N} \times \mathbb{N} \to [0,1]$ and argument soundness $\beta\colon \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to [0,1]$.*

There is no relativized argument $(\mathcal{P}, \mathcal{V})$ for the relativized relation $R_{\mathcal{O}}$ with completeness error α, soundness error β, argument verifier running time vt, argument size as, argument honest prover query bound pq, and argument honest prover time bound pt if there exists $\lambda \in \mathbb{N}$, $n \in \mathbb{N}$ and $p\colon \mathbb{N} \to [0,1]$ such that the following holds:

- $0 \leq \beta(\lambda, n, \mathsf{pq}, \mathsf{pt}) < \left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}}$,
- $0 \leq \frac{2^{t(n)}-1}{2^{t(n)}} \cdot \alpha(\lambda, n) < (1 - t(n) \cdot p(n)) \cdot \frac{2^{t(n)} - \frac{2^{O(\mathsf{as}(\lambda,n))} \cdot \mathsf{vq}(\lambda,n)}{p(n)}}{2^{t(n)}} \cdot \left(\left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}} - \beta(\lambda, n, \mathsf{pq}, \mathsf{pt})\right)$, *and*
- $(1 - t(n) \cdot p(n)) \cdot \frac{2^{t(n)} - \frac{2^{O(\mathsf{as}(\lambda,n))} \cdot \mathsf{vq}(\lambda,n)}{p(n)}}{2^{t(n)}} \cdot \left(\left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}} - \beta(\lambda, n, \mathsf{pq}, \mathsf{pt})\right) \leq 1.$

We prove Lemma 5.4 in Section 5.2 and Lemma 5.5 in Section 5.3.

5.2 Proof of Lemma 5.4

To show that the relation $R_{\mathcal{O}}$ is in $\mathsf{NTIME}[t]$, we construct a nondeterministic Turing machine M that decides $R_{\mathcal{O}}$ within $O(t(n))$ time.

$M^f(\mathbb{x}, \mathbb{w})$:
1. If $\mathbb{x} \neq 0^n$, reject.
2. For every $i \in [t(n)]$:
   (a) If $f(\mathbb{w} \parallel u_{t(n),i})_1 = 1$, reject.
3. Accept.

It is easy to argue that for every $\lambda \in \mathbb{N}$ and $f \in \mathcal{O}_\lambda$, $M^f(\mathbb{x}, \mathbb{w}) = 1$ if and only if $(\mathbb{x}, \mathbb{w}) \in R_f$.

Now we argue that M^f decides R_f within time $t(n)$. Checking whether $\mathbb{x} = 0^n$ takes $O(n)$ time. Each query to f incurs a time cost of $O(1)$, and M makes $t(n)$ such invocations. Writing the initial query tape necessitates $O(t(n))$ steps, and each subsequent query tape update requires amortized $O(1)$ steps. Thus, M decides the relation $R_{\mathcal{O}}$ in $O(t(n))$ time.

5.3 Proof of Lemma 5.5

Assume for the sake of contradiction that $R_{\mathcal{O}}$ has such a relativized argument system $(\mathcal{P},\mathcal{V})$ and there exists $\lambda \in \mathbb{N}$, $n \in \mathbb{N}$ and $p \colon \mathbb{N} \to [0,1]$ that satisfy the conditions in Lemma 5.5.

For every $f \in \mathcal{O}_\lambda$, we define the set $Q^\star(f)$ as the following set:

$$\left\{ w \in \{0,1\}^{t(n)} \;\middle|\; \forall i \in [t(n)], \sum_{\pi \in \{0,1\}^{\leq \mathsf{as}}} \Pr\left[\mathcal{V}^f(1^\lambda, 0^n, \pi, \rho) \text{ queries } f \text{ at } w \parallel u_{t(n),i} \;\middle|\; \rho \leftarrow \{0,1\}^{\mathsf{vr}} \right] < p(n) \right\} \ .$$

Intuitively, $Q^\star(f)$ is the set of strings $w \in \{0,1\}^{t(n)}$ for which $\mathcal{V}^f$ queries all of $\{w \parallel u_{t(n),i}\}_{i\in[t(n)]}$ with *low* probability.

We first argue that size of $Q^\star(f)$ is large. Consider the set $Q_c(f)$ of string $w \in \{0,1\}^{t(n)}$ that $\mathcal{V}^f$ queries with *high* probability:

$$\left\{ w \parallel u \in \{0,1\}^{t(n)+\lceil \log t(n) \rceil} \;\middle|\; \sum_{\pi \in \{0,1\}^{\leq \mathsf{as}}} \Pr\left[\mathcal{V}^f(1^\lambda, 0^n, \pi, \rho) \text{ queries } f \text{ at } w \parallel u \;\middle|\; \rho \leftarrow \{0,1\}^{\mathsf{vr}} \right] \geq p(n) \right\} \ .$$

Since $\mathcal{V}^f$ can make at most $\mathsf{vq}(\lambda,n)$ queries to f and $|\pi| \leq \mathsf{as}(\lambda,n)$,

$$\sum_{w\parallel u \in \{0,1\}^{t(n)+\lceil \log t(n)\rceil}} \sum_{\pi \in \{0,1\}^{\leq \mathsf{as}}} \Pr\left[\mathcal{V}^f(1^\lambda, 0^n, \pi, \rho) \text{ queries } f \text{ at } w \parallel u \;\middle|\; \rho \leftarrow \{0,1\}^{\mathsf{vr}} \right] \leq 2^{O(\mathsf{as}(\lambda,n))} \cdot \mathsf{vq}(\lambda,n) \ .$$

Hence,

$$|Q_c(f)| \leq \frac{2^{O(\mathsf{as}(\lambda,n))} \cdot \mathsf{vq}(\lambda,n)}{p(n)} \ .$$

Therefore,

$$|Q^\star(f)| \geq 2^{t(n)} - |Q_c(f)| \geq 2^{t(n)} - \frac{2^{O(\mathsf{as}(\lambda,n))} \cdot \mathsf{vq}(\lambda,n)}{p(n)} \ . \tag{5}$$

Moreover, by definition of $Q^\star(f)$, we can say that for every $w \in Q^\star(f)$,

$$\forall \pi \in \{0,1\}^{\leq \mathsf{as}}, \forall i \in [t(n)], \Pr\left[\mathcal{V}^f(1^\lambda, 0^n, \pi, \rho) \text{ queries } f \text{ at } w \parallel u_{t(n),i} \;\middle|\; \rho \leftarrow \{0,1\}^{\mathsf{vr}} \right] < p(n) \ .$$

We can deduce that

$$\begin{aligned}
\forall \pi \in \{0,1\}^{\leq \mathsf{as}}, \Pr&\left[\exists i \in [t(n)], \mathcal{V}^f(1^\lambda, 0^n, \pi, \rho) \text{ queries } f \text{ at } w \parallel u_{t(n),i} \;\middle|\; \rho \leftarrow \{0,1\}^{\mathsf{vr}} \right] \\
&\leq \sum_{i \in [t(n)]} \Pr\left[\mathcal{V}^f(1^\lambda, 0^n, \pi, \rho) \text{ queries } f \text{ at } w \parallel u_{t(n),i} \;\middle|\; \rho \leftarrow \{0,1\}^{\mathsf{vr}} \right] \\
&< t(n) \cdot p(n) \ ,
\end{aligned}$$

which implies that for every $\pi \in \{0,1\}^{\leq \mathsf{as}}$,

$$\Pr\left[\forall i \in [t(n)], \mathcal{V}^f(1^\lambda, 0^n, \pi, \rho) \text{ doesn't query } f \text{ at } w \parallel u_{t(n),i} \;\middle|\; \rho \leftarrow \{0,1\}^{\mathsf{vr}} \right] \geq 1 - t(n) \cdot p(n) \ . \tag{6}$$

For every $f \in \mathcal{O}_\lambda$ and $w \in \{0,1\}^{t(n)}$, we define the operation $\mathsf{SetZero}(f,w)$ as follows:

$\mathsf{SetZero}(f, w)$:
1. Initialize $f' := f$.
2. For all $i \in [t(n)]$: $f'(w \parallel u_{t(n),i})_1 := 0$.
3. Output f'.

For every $w \in \{0,1\}^{t(n)}$, we define the following malicious prover $\widetilde{\mathcal{P}}_{n,w}$:

$\widetilde{\mathcal{P}}^f_{n,w}$:
1. Set $(\mathbb{x}, \mathbb{w}) := (0^n, w)$.
2. Sample the randomness for honest argument prover: $\zeta \leftarrow \{0,1\}^{\mathsf{pr}}$.
3. Simulate $\mathcal{P}(1^\lambda, (\mathbb{x}, \mathbb{w}), \zeta)$.
4. For any $i \in [t(n)]$, if $\mathcal{P}$ makes a query to $w \parallel u_{t(n),i}$:
    (a) Let $\mathsf{ans} := f(w \parallel u_{t(n),i})$.
    (b) Set $\mathsf{ans}' := \mathsf{ans}$.
    (c) Set $\mathsf{ans}'_1 := 0$.
    (d) Give ans' as answer to the query $w \parallel u_{t(n),i}$.
5. Let $\widetilde{\pi}'$ be the output of $\mathcal{P}(1^\lambda, (\mathbb{x}, \mathbb{w}), \zeta)$ with the altered answer.
6. Output $(\mathbb{x}, \widetilde{\pi}')$.

Note that $\widetilde{\mathcal{P}}$ has the same query complexity and running time as $\mathcal{P}$. Moreover, for every argument prover randomness $\zeta \in \{0,1\}^{\mathsf{pr}}$ and $w \in \{0,1\}^{t(n)}$, $(0^n, \mathcal{P}^{\mathsf{SetZero}(f,w)}(1^\lambda, (0^n, w), \zeta)) = \widetilde{\mathcal{P}}^f_{n,w}(\zeta)$. We also define the completeness adversary $\mathcal{A}_{n,w}$ for every $w \in \{0,1\}^n$:

$\mathcal{A}^f_{n,w}$: Output $(\mathbb{x}, \mathbb{w}) := (0^n, w)$.

Moreover, we define $\mathcal{A}_n$ as follows:

$\mathcal{A}^f_n$:
1. Let $(\mathbb{x}, \mathbb{w}) := (0^n, 0^{t(n)})$.
2. If there exists $w \in \{0,1\}^{t(n)}$ such that for all $i \in [t(n)]$, $f(w \parallel u_{t(n),i})_1 = 0$, set $\mathbb{w} := w$.
3. Output $(\mathbb{x}, \mathbb{w})$.

For the derivation below, we omit the sampling of the verifier's randomness ρ and prover's randomness ζ in the experiment, they are always sampled as follows:

$$\left[\begin{array}{c} \zeta \leftarrow \{0,1\}^{\mathsf{pr}} \\ \rho \leftarrow \{0,1\}^{\mathsf{vr}} \end{array}\right] .$$

Moreover, for every $f \in \mathcal{O}_\lambda$ and $w \in \{0,1\}^{t(n)}$, we define the predicate $\phi(f, w)$ as follows:

$$\phi(f,w) := \begin{cases} 1 & \text{if } \forall i \in [t(n)], f(w \parallel u_{t(n),i})_1 = 0 \\ 0 & \text{otherwise} \end{cases} .$$

$$\Pr\left[\begin{array}{l} 0^n \notin L(R_f) \\ \wedge \mathcal{V}^{f'}(1^\lambda, 0^n, \pi, \rho) = 0 \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ w \leftarrow \{0,1\}^{t(n)} \\ f' := \mathsf{SetZero}(f, w) \\ (0^n, w) := \mathcal{A}^f_{n,w} \\ \pi := \mathcal{P}^{f'}(1^\lambda, (0^n, w), \zeta) \end{array}\right]$$

$$
\begin{aligned}
&= \sum_{\substack{f^\star \in \mathcal{O}_\lambda \\ 0^n \notin L(R_{f^\star}) \\ w^\star \in \{0,1\}^{t(n)} \\ f' := \mathsf{SetZero}(f^\star, w^\star)}} \Pr\left[\begin{array}{c} f = f^\star \\ \wedge\, w = w^\star \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ w \leftarrow \{0,1\}^{t(n)} \end{array}\right] \cdot \Pr\left[\mathcal{V}^{f'}(1^\lambda, 0^n, \pi, \rho) = 0 \middle| \begin{array}{l} (0^n, w^\star) := \mathcal{A}^{f^\star}_{n,w^\star} \\ \pi := \mathcal{P}^{f'}(1^\lambda, (0^n, w^\star), \zeta) \end{array}\right] \\
&= \frac{1}{2^{t(n)}} \cdot \sum_{\substack{f^\star \in \mathcal{O}_\lambda \\ 0^n \notin L(R_{f^\star}) \\ w^\star \in \{0,1\}^{t(n)} \\ f' := \mathsf{SetZero}(f^\star, w^\star)}} \Pr\left[f = f^\star \mid f \leftarrow \mathcal{O}_\lambda\right] \cdot \Pr\left[\mathcal{V}^{f'}(1^\lambda, 0^n, \pi, \rho) = 0 \middle| \begin{array}{l} (0^n, w^\star) := \mathcal{A}^{f^\star}_{n,w^\star} \\ \pi := \mathcal{P}^{f'}(1^\lambda, (0^n, w^\star), \zeta) \end{array}\right] \\
&= \frac{1}{2^{t(n)}} \sum_{\substack{w^\star \in \{0,1\}^{t(n)} \\ f^\star \in \mathcal{O}_\lambda \\ 0^n \notin L(R_{f^\star}) \\ f' := \mathsf{SetZero}(f^\star, w^\star)}} \Pr\left[f = f^\star \mid f \leftarrow \mathcal{O}_\lambda\right] \cdot \Pr\left[\mathcal{V}^{f'}(1^\lambda, 0^n, \pi, \rho) = 0 \middle| \begin{array}{l} (0^n, w^\star) := \mathcal{A}^{f^\star}_{n,w^\star} \\ \pi := \mathcal{P}^{f'}(1^\lambda, (0^n, w^\star), \zeta) \end{array}\right] \\
&=_{[1]} \frac{2^{t(n)} - 1}{2^{t(n)}} \sum_{\substack{w^\star \in \{0,1\}^{t(n)} \\ f' \in \mathcal{O}_\lambda \\ \phi(f', w^\star) = 1 \\ \forall w \neq w^\star, \phi(f', w) = 0}} \Pr\left[f = f' \mid f \leftarrow \mathcal{O}_\lambda\right] \cdot \Pr\left[\mathcal{V}^{f'}(1^\lambda, 0^n, \pi, \rho) = 0 \middle| \begin{array}{l} (0^n, w^\star) := \mathcal{A}^{f'}_{n,w^\star} \\ \pi := \mathcal{P}^{f'}(1^\lambda, (0^n, w^\star), \zeta) \end{array}\right] \\
&= \frac{2^{t(n)} - 1}{2^{t(n)}} \sum_{\substack{w^\star \in \{0,1\}^{t(n)} \\ f^\star \in \mathcal{O}_\lambda \\ \phi(f^\star, w^\star) = 1 \\ \forall w \neq w^\star, \phi(f^\star, w) = 0}} \Pr\left[f = f^\star \mid f \leftarrow \mathcal{O}_\lambda\right] \cdot \Pr\left[\mathcal{V}^{f^\star}(1^\lambda, 0^n, \pi, \rho) = 0 \middle| \begin{array}{l} (0^n, w^\star) := \mathcal{A}^{f^\star}_{n,w^\star} \\ \pi := \mathcal{P}^{f^\star}(1^\lambda, (0^n, w^\star), \zeta) \end{array}\right] \\
&= \frac{2^{t(n)} - 1}{2^{t(n)}} \sum_{\substack{f^\star \in \mathcal{O}_\lambda \\ \exists! w \in \{0,1\}^{t(n)}, \phi(f^\star, w) = 1}} \Pr\left[f = f^\star \mid f \leftarrow \mathcal{O}_\lambda\right] \cdot \Pr\left[\mathcal{V}^{f^\star}(1^\lambda, 0^n, \pi, \rho) = 0 \middle| \begin{array}{l} (0^n, w) := \mathcal{A}^{f^\star}_{n} \\ \pi := \mathcal{P}^{f^\star}(1^\lambda, (0^n, w), \zeta) \end{array}\right] \\
&\leq \frac{2^{t(n)} - 1}{2^{t(n)}} \sum_{\substack{f^\star \in \mathcal{O}_\lambda \\ \exists w \in \{0,1\}^{t(n)}, \phi(f^\star, w) = 1}} \Pr\left[f = f^\star \mid f \leftarrow \mathcal{O}_\lambda\right] \cdot \Pr\left[\mathcal{V}^{f^\star}(1^\lambda, 0^n, \pi, \rho) = 0 \middle| \begin{array}{l} (0^n, w) := \mathcal{A}^{f^\star}_{n} \\ \pi := \mathcal{P}^{f^\star}(1^\lambda, (0^n, w), \zeta) \end{array}\right] \\
&=_{[2]} \frac{2^{t(n)} - 1}{2^{t(n)}} \sum_{\substack{f^\star \in \mathcal{O}_\lambda \\ \exists w \in \{0,1\}^{t(n)}, (0^n, w) \in R_{f^\star}}} \Pr\left[f = f^\star \mid f \leftarrow \mathcal{O}_\lambda\right] \cdot \Pr\left[\mathcal{V}^{f^\star}(1^\lambda, 0^n, \pi, \rho) = 0 \middle| \begin{array}{l} (0^n, w) := \mathcal{A}^{f}_{n} \\ \pi := \mathcal{P}^{f^\star}(1^\lambda, (0^n, w), \zeta) \end{array}\right] \\
&= \frac{2^{t(n)} - 1}{2^{t(n)}} \cdot \Pr\left[\begin{array}{l} (0^n, w) \in R_f \\ \wedge\, \mathcal{V}^f(1^\lambda, 0^n, \pi, \rho) = 0 \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, w) := \mathcal{A}^f_n \\ \pi := \mathcal{P}^f(1^\lambda, (0^n, w), \zeta) \end{array}\right] \\
&\leq_{[3]} \frac{2^{t(n)} - 1}{2^{t(n)}} \cdot \alpha(\lambda, n) \ ,
\end{aligned}
$$

where

- Equality [1]: By definition of the relation $R_{\mathcal{O}}$ (Definition 5.3), for every $f \in \mathcal{O}_\lambda$ such that $0^n \notin L_f$, for every $w \in \{0,1\}^n$, there exists some $i \in [t(n)]$ such that $f(w \parallel u_{t(n),i})_1 = 1$. For every $w \in \{0,1\}^n$, let

$f_w \in \mathcal{O}_\lambda$ be a function where w is the only witness such that $f_w(w \parallel u_{t(n),i})_1 = 0$ for all $i \in [t(n)]$. There are exactly $2^{t(n)} - 1$ many $f \in \mathcal{O}_\lambda$ where $0^n \notin L_f$ such that $\mathsf{SetZero}(f, w) = f_w$;

- Equality [2] follows from definition of $R_{\mathcal{O}}$ in Definition 5.3;
- Inequality [3] follows from Definition 3.12.

On the other hand, we can also obtain the following:

$$
\begin{aligned}
&\Pr\left[\begin{array}{l} 0^n \notin L(R_f) \\ \wedge \mathcal{V}^{f'}(1^\lambda, 0^n, \pi, \rho) = 0 \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ w \leftarrow \{0,1\}^{t(n)} \\ f' := \mathsf{SetZero}(f, w) \\ (0^n, w) := \mathcal{A}^f_{n,w} \\ \pi := \mathcal{P}^{f'}(1^\lambda, (0^n, w), \zeta) \end{array}\right] \\
\geq_{[1]} &\Pr\left[\begin{array}{l} \mathcal{V}^{f'}(\mathbb{x}, \pi, \rho) = 0 \\ \text{conditioned on} \\ w \in Q^\star(f) \\ \wedge 0^n \notin L(R_f) \\ \wedge \mathcal{V}^f(1^\lambda, 0^n, \widetilde{\pi}, \rho) = 0 \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ w \leftarrow \{0,1\}^{t(n)} \\ f' := \mathsf{SetZero}(f, w) \\ (0^n, w) := \mathcal{A}^f_{n,w} \\ \pi := \mathcal{P}^{f'}(1^\lambda, (0^n, w), \zeta) \\ (0^n, \widetilde{\pi}) := \widetilde{\mathcal{P}}^f_{n,w}(\zeta) \end{array}\right] \cdot \Pr\left[\begin{array}{l} w \in Q^\star(f) \\ \wedge 0^n \notin L(R_f) \\ \wedge \mathcal{V}^f(1^\lambda, 0^n, \widetilde{\pi}, \rho) = 0 \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ w \leftarrow \{0,1\}^{t(n)} \\ (0^n, \widetilde{\pi}) := \widetilde{\mathcal{P}}^f_{n,w}(\zeta) \end{array}\right] \\
\geq_{[2]} &(1 - t(n) \cdot p(n)) \cdot \Pr\left[\begin{array}{l} w \in Q^\star(f) \\ \wedge 0^n \notin L(R_f) \\ \wedge \mathcal{V}^f(1^\lambda, 0^n, \widetilde{\pi}, \rho) = 0 \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ w \leftarrow \{0,1\}^{t(n)} \\ (0^n, \widetilde{\pi}) := \widetilde{\mathcal{P}}^f_{n,w}(\zeta) \end{array}\right] ,
\end{aligned}
$$

where

- Inequality [1] follows from definition of conditional probability;
- Inequality [2] follows from Eq. (6) and that for every $f \in \mathcal{O}_\lambda$, $w \in \{0,1\}^{t(n)}$, and $\zeta \in \{0,1\}^{\mathsf{pr}}$, $\widetilde{\mathcal{P}}^f_{n,w}(\zeta)$ and $\mathcal{P}^{\mathsf{SetZero}(f,w)}(1^\lambda, (0^n, w), \zeta)$ output the same proof.

Moreover,

$$
\begin{aligned}
&\Pr\left[\begin{array}{l} w \in Q^\star(f) \\ \wedge 0^n \notin L(R_f) \\ \wedge \mathcal{V}^f(1^\lambda, 0^n, \widetilde{\pi}, \rho) = 0 \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ w \leftarrow \{0,1\}^{t(n)} \\ (0^n, \widetilde{\pi}) := \widetilde{\mathcal{P}}^f_{n,w}(\zeta) \end{array}\right] \\
= &\sum_{\substack{w^\star \in \{0,1\}^{t(n)} \\ f^\star \in \mathcal{O}_\lambda \\ w^\star \in Q^\star(f^\star)}} \Pr\left[\begin{array}{l} w = w^\star \\ \wedge f = f^\star \end{array} \middle| \begin{array}{l} w \leftarrow \{0,1\}^{t(n)} \\ f \leftarrow \mathcal{O}_\lambda \end{array}\right] \cdot \Pr\left[\begin{array}{l} 0^n \notin L(R_{f^\star}) \\ \wedge \mathcal{V}^{f^\star}(1^\lambda, 0^n, \widetilde{\pi}, \rho) = 0 \end{array} \middle| (0^n, \widetilde{\pi}) := \widetilde{\mathcal{P}}^{f^\star}_{n,w^\star}(\zeta)\right] \\
= &\frac{1}{2^{t(n)}} \sum_{w^\star \in \{0,1\}^{t(n)}} \sum_{\substack{f^\star \in \mathcal{O}_\lambda \\ w^\star \in Q^\star(f^\star)}} \Pr[f = f^\star \mid f \leftarrow \mathcal{O}_\lambda] \cdot \Pr\left[\begin{array}{l} 0^n \notin L(R_{f^\star}) \\ \wedge \mathcal{V}^{f^\star}(1^\lambda, 0^n, \widetilde{\pi}, \rho) = 0 \end{array} \middle| (0^n, \widetilde{\pi}) := \widetilde{\mathcal{P}}^{f^\star}_{n,w^\star}(\zeta)\right] \\
\geq_{[1]} &\frac{2^{t(n)} - \frac{2^{O(\mathsf{as}(\lambda,n))} \cdot \mathsf{vq}(\lambda,n)}{p(n)}}{2^{t(n)}}
\end{aligned}
$$

$$
\begin{aligned}
&\cdot \min_{w^\star \in \{0,1\}^{t(n)}} \left\{ \sum_{f^\star \in \mathcal{O}_\lambda} \Pr\left[f = f^\star \mid f \leftarrow \mathcal{O}_\lambda\right] \cdot \Pr\left[\begin{array}{l} 0^n \notin L(R_{f^\star}) \\ \wedge\, \mathcal{V}^{f^\star}(1^\lambda, 0^n, \widetilde{\pi}, \rho) = 0 \end{array} \;\middle|\; (0^n, \widetilde{\pi}) \coloneqq \widetilde{\mathcal{P}}^{f^\star}_{n,w^\star}(\zeta) \right] \right\} \\
&= \frac{2^{t(n)} - \frac{2^{O(\mathsf{as}(\lambda,n))} \cdot \mathsf{vq}(\lambda,n)}{p(n)}}{2^{t(n)}} \cdot \min_{w^\star \in \{0,1\}^{t(n)}} \left\{ \Pr\left[\begin{array}{l} 0^n \notin L(R_f) \\ \wedge\, \mathcal{V}^{f}(1^\lambda, 0^n, \widetilde{\pi}, \rho) = 0 \end{array} \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, \widetilde{\pi}) \coloneqq \widetilde{\mathcal{P}}^{f}_{n,w^\star}(\zeta) \end{array} \right] \right\} \\
&=_{[2]} \frac{2^{t(n)} - \frac{2^{O(\mathsf{as}(\lambda,n))} \cdot \mathsf{vq}(\lambda,n)}{p(n)}}{2^{t(n)}} \\
&\quad \cdot \min_{w^\star \in \{0,1\}^{t(n)}} \left\{ \Pr\left[0^n \notin L(R_f) \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, \widetilde{\pi}) \coloneqq \widetilde{\mathcal{P}}^{f}_{n,w^\star}(\zeta) \end{array} \right] - \Pr\left[\begin{array}{l} 0^n \notin L(R_f) \\ \wedge\, \mathcal{V}^{f}(1^\lambda, 0^n, \widetilde{\pi}, \rho) = 1 \end{array} \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, \widetilde{\pi}) \coloneqq \widetilde{\mathcal{P}}^{f}_{n,w^\star}(\zeta) \end{array} \right] \right\} \\
&=_{[3]} \frac{2^{t(n)} - \frac{2^{O(\mathsf{as}(\lambda,n))} \cdot \mathsf{vq}(\lambda,n)}{p(n)}}{2^{t(n)}} \cdot \min_{w^\star \in \{0,1\}^{t(n)}} \left\{ \left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}} - \Pr\left[\begin{array}{l} 0^n \notin L(R_f) \\ \wedge\, \mathcal{V}^{f}(1^\lambda, 0^n, \widetilde{\pi}, \rho) = 1 \end{array} \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, \widetilde{\pi}) \coloneqq \widetilde{\mathcal{P}}^{f}_{n,w^\star}(\zeta) \end{array} \right] \right\} \\
&\geq_{[4]} \frac{2^{t(n)} - \frac{2^{O(\mathsf{as}(\lambda,n))} \cdot \mathsf{vq}(\lambda,n)}{p(n)}}{2^{t(n)}} \cdot \left(\left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}} - \beta(\lambda, n, \mathsf{pq}, \mathsf{pt}) \right) \quad ,
\end{aligned}
$$

where

- Inequality [1] follows from Eq. (5);
- Equality [2] follows from the law of total probability;
- Equality [3] holds: Let $F_{n,0} \coloneqq \{f \in \mathcal{O}_\lambda : 0^n \notin L(R_f)\}$, then the measure $\mu_{\mathcal{O}_\lambda}(F_{n,0}) = \left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}}$ because the probability measure of $\mathcal{O}_\lambda$ is uniform;
- Inequality [4] follows from Definition 3.13.

We can conclude that

$$
(1 - t(n) \cdot p(n)) \cdot \frac{2^{t(n)} - \frac{2^{O(\mathsf{as}(\lambda,n))} \cdot \mathsf{vq}(\lambda,n)}{p(n)}}{2^{t(n)}} \cdot \left(\left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}} - \beta(\lambda, n, \mathsf{pq}, \mathsf{pt}) \right) \leq \frac{2^{t(n)} - 1}{2^{t(n)}} \cdot \alpha(\lambda, n) \quad ,
$$

a contradiction to our assumption.

6 Alternative separation of NTIME and NARG

Theorem 6.1. *Let $\mathcal{O}$ be the random oracle. For every time-constructible function $t\colon \mathbb{N} \to \mathbb{N}$, argument verifier query bound $\mathsf{vq}\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}$, argument verifier time bound $\mathsf{vt}\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}$, argument size bound $\mathsf{as}\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}$, argument honest prover query bound $\mathsf{pq}\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}$, argument honest prover time bound $\mathsf{pt}\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}$, argument completeness $\alpha\colon \mathbb{N}\times\mathbb{N} \to [0,1]$ and argument soundness $\beta\colon \mathbb{N}\times\mathbb{N}\times\mathbb{N}\times\mathbb{N} \to [0,1]$,*

$$\mathsf{NTIME}^{\mathcal{O}}\left[t(n)\right] \not\subseteq \mathsf{NARG}^{\mathcal{O}}\begin{bmatrix} \text{completeness error} & \alpha & = & \alpha(\lambda,n) \\ \text{soundness error} & \beta & = & \beta(\lambda,n,\mathsf{q}_{\widetilde{\mathcal{P}}},\mathsf{t}_{\widetilde{\mathcal{P}}}) \\ \text{verifier query bound} & \mathsf{vq} & = & \mathsf{vq}(\lambda,n) \\ \text{verifier running time} & \mathsf{vt} & = & \mathsf{vt}(\lambda,n) \\ \text{argument size} & \mathsf{as} & = & \mathsf{as}(\lambda,n) \\ \text{honest prover query bound} & \mathsf{pq} & = & \mathsf{pq}(\lambda,n) \\ \text{honest prover time bound} & \mathsf{pt} & = & \mathsf{pt}(\lambda,n) \end{bmatrix},$$

where there exists $\lambda \in \mathbb{N}$, $n \in \mathbb{N}$ and some function $p\colon \mathbb{N} \to (0,1]$ such that the following holds:

- $0 \le \alpha(\lambda,n) < \left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}-1}$,
- $0 \le \beta(\lambda,n,\mathsf{pq},\mathsf{pt}) < (1-p(n)) \cdot (t(n) - \frac{\mathsf{vq}(\lambda,n)}{p(n)}) \cdot \left(\left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}-1} - \alpha(\lambda,n)\right)$, *and*
- $(1-p(n)) \cdot \frac{t(n) - \frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)} \cdot \left(\left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}-1} - \alpha(\lambda,n)\right) \le 1.$

The corollary below follows by setting $p(n) < \frac{1}{2}$ for all $n \in \mathbb{N}$ in Theorem 6.1.

Corollary 6.2. *Let $\mathcal{O}$ be the random oracle. For every time-constructible function $t\colon \mathbb{N}\times\mathbb{N}$, $\lambda \in \mathbb{N}$ and $n \in \mathbb{N}$,*

$$\mathsf{NTIME}^{\mathcal{O}}\left[t(n)\right] \not\subseteq \mathsf{NARG}^{\mathcal{O}}\begin{bmatrix} \text{completeness error} & \alpha & = & 1/3 \\ \text{soundness error} & \beta & = & \begin{cases} 1/3 & \mathsf{q}_{\widetilde{\mathcal{P}}} = \mathsf{poly}(\lambda,t(n)), \mathsf{t}_{\widetilde{\mathcal{P}}} = \mathsf{poly}(\lambda,t(n)) \\ 1 & \textit{otherwise} \end{cases} \\ \text{verifier query bound} & \mathsf{vq} & = & o(t(n)) \\ \text{verifier running time} & \mathsf{vt} & & \\ \text{argument size} & \mathsf{as} & & \\ \text{honest prover query bound} & \mathsf{pq} & = & \mathsf{poly}(\lambda,t(n)) \\ \text{honest prover time bound} & \mathsf{pq} & = & \mathsf{poly}(\lambda,t(n)) \end{bmatrix}.$$

6.1 Separation for every security parameter

Similar to Section 5, to prove Theorem 6.1, it is enough to find one relativized relation $R_{\mathcal{O}}$ that can be decided by an oracle nondeterministic Turing machine within $O(t(n))$ time but does not have a relativized argument system. In fact, we consider the same relativized relation $R_{\mathcal{O}}$ defined in Definition 5.3.

The following lemma and Lemma 5.4 directly imply Theorem 6.1.

Lemma 6.3. *Let $\mathcal{O}$ be the random oracle. Let $R_{\mathcal{O}}$ be defined as in Definition 5.3. Fix time-constructible function $t\colon \mathbb{N} \to \mathbb{N}$, argument verifier query bound $\mathsf{vq}\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}$, argument verifier time bound $\mathsf{vt}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument size bound $\mathsf{as}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument honest prover query bound $\mathsf{pq}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument honest prover time bound $\mathsf{pt}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument completeness $\alpha\colon \mathbb{N}\times\mathbb{N}\to[0,1]$ and argument soundness $\beta\colon \mathbb{N}\times\mathbb{N}\times\mathbb{N}\times\mathbb{N}\to[0,1]$.*

There is no relativized argument $(\mathcal{P},\mathcal{V})$ for the relativized relation $R_{\mathcal{O}}$ with completeness error α, soundness error β, argument verifier running time vt, argument size as, argument honest prover query bound pq, and argument honest prover time bound pt if there exists $\lambda\in\mathbb{N}$, $n\in\mathbb{N}$ and $p\colon\mathbb{N}\to[0,1]$ such that the following holds:

- $0 \le \alpha(\lambda,n) < \left(1-\frac{1}{2^{t(n)}}\right)^{2^{t(n)}-1}$,
- $0 \le \beta(\lambda,n,\mathsf{pq},\mathsf{pt}) < (1-p(n))\cdot(t(n)-\frac{\mathsf{vq}(\lambda,n)}{p(n)})\cdot\left(\left(1-\frac{1}{2^{t(n)}}\right)^{2^{t(n)}-1}-\alpha(\lambda,n)\right)$, *and*
- $(1-p(n))\cdot\frac{t(n)-\frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)}\cdot\left(\left(1-\frac{1}{2^{t(n)}}\right)^{2^{t(n)}-1}-\alpha(\lambda,n)\right)\le 1.$

We prove Lemma 6.3 in Section 6.2.

6.2 Proof of Lemma 6.3

Assume for the sake of contradiction that $R_{\mathcal{O}}$ has such a relativized argument system $(\mathcal{P},\mathcal{V})$ and there exists $\lambda\in\mathbb{N}$, $n\in\mathbb{N}$ and $p\colon\mathbb{N}\to[0,1]$ that satisfy the conditions in Lemma 6.3.

We define the following sets:

- $\mathsf{F}_{n,1} \coloneqq \{f\in\mathcal{O}_\lambda : 0^n\in L(R_f)\}$;
- $\mathsf{F}_{n,0} \coloneqq \{f\in\mathcal{O}_\lambda : 0^n\notin L(R_f)\}$;
- $\mathsf{UF}_n \coloneqq \left\{f\in\mathcal{O}_\lambda : \exists! w\in\{0,1\}^{t(n)}, (0^n,w)\in R_f\right\}$;
- For every $w\in\{0,1\}^{t(n)}$, $\mathsf{UF}_{n,w}\coloneqq\{f\in\mathcal{O}_\lambda : f\in\mathsf{UF}_n\wedge(0^n,w)\in R_f\}$.

For every $f\in\mathcal{O}_\lambda$ and $w\in\{0,1\}^{t(n)}$, we define the set $Q^\star(f,w)$ as the following set:

$$\left\{i\in[t(n)]\;\middle|\;\Pr\left[\mathcal{V}^f(1^\lambda,0^n,\pi,\rho)\text{ queries }f\text{ at }w\parallel u_{t(n),i}\;\middle|\;\begin{array}{c}\pi\leftarrow\mathcal{P}^f(0^n,w)\\ \rho\leftarrow\{0,1\}^{\mathsf{vr}}\end{array}\right]<p(n)\right\}\ .$$

Consider the set $Q_c(f,w)$ that $\mathcal{V}^f$ queries with *high* probability:

$$\left\{i\in[t(n)]\;\middle|\;\Pr\left[\mathcal{V}^f(1^\lambda,0^n,\pi,\rho)\text{ queries }f\text{ at }w\parallel u_{t(n),i}\;\middle|\;\begin{array}{c}\pi\leftarrow\mathcal{P}^f(0^n,w)\\ \rho\leftarrow\{0,1\}^{\mathsf{vr}}\end{array}\right]\ge p(n)\right\}\ .$$

Since $\mathcal{V}^f$ makes at most $\mathsf{vq}(\lambda,n)$ queries,

$$\sum_{i\in[t(n)]}\Pr\left[\mathcal{V}^f(1^\lambda,0^n,\pi,\rho)\text{ queries }f\text{ at }w\parallel u_{t(n),i}\;\middle|\;\begin{array}{c}\pi\leftarrow\mathcal{P}^f(0^n,w)\\ \rho\leftarrow\{0,1\}^{\mathsf{vr}}\end{array}\right]\le\mathsf{vq}(\lambda,n)\ .$$

Hence,

$$|Q_c(f,w)| \leq \frac{\mathsf{vq}(\lambda,n)}{p(n)} \ .$$

Therefore,

$$|Q^\star(f,w)| = t(n) - |Q_c(f,w)| \geq t(n) - \frac{\mathsf{vq}(\lambda,n)}{p(n)} \ . \tag{7}$$

Moreover, by definition of $Q^\star(f,w)$, we can say that for every $i \in Q^\star(f,w)$,

$$\Pr\left[\mathcal{V}^f(1^\lambda, 0^n, \pi, \rho) \text{ doesn't query } f \text{ at } w \parallel u_{t(n),i} \;\middle|\; \begin{array}{c} \pi \leftarrow \mathcal{P}^f(0^n, w) \\ \rho \leftarrow \{0,1\}^{\mathsf{vr}} \end{array}\right] \geq 1 - p(n) \ . \tag{8}$$

For every $f \in \mathcal{O}_\lambda$, $w \in \{0,1\}^{t(n)}$, and $i \in [t(n)]$, we define the operation $\mathsf{Flip}\,(f,w,i)$ as follows:

$\mathsf{Flip}\,(f,w,i)$:
1. Initialize $f' := f$.
2. Set $f'(w \parallel u_{t(n),i})_1 := 1 - f(w \parallel u_{t(n),i})_1$.
3. Output f'.

We define the completeness adversary $\mathcal{A}_{n,w}$ as follows:

$\mathcal{A}^f_{n,w}$: Output $(0^n, w)$.

Moreover, we define $\mathcal{A}_n$ as follows:

$\mathcal{A}^f_n$:
1. Let $(\mathbbm{x}, \mathbbm{w}) := (0^n, 0^{t(n)})$.
2. If there exists $w \in \{0,1\}^{t(n)}$ such that for all $i \in [t(n)]$, $f(w \parallel u_{t(n),i})_1 = 0$, set $\mathbbm{w} := w$.
3. Output $(\mathbbm{x}, \mathbbm{w})$.

We define the argument adversary $\widetilde{\mathcal{P}}_{n,w,i}$:

$\widetilde{\mathcal{P}}^f_{n,w,i}$:
1. Set $(\mathbbm{x}, \mathbbm{w}) := (0^n, w)$.
2. Sample the randomness for the honest argument prover: $\zeta \leftarrow \{0,1\}^{\mathsf{pr}}$.
3. Simulate $\mathcal{P}(1^\lambda, (\mathbbm{x}, \mathbbm{w}), \zeta)$.
4. If $\mathcal{P}$ makes a query to $w \parallel u_{t(n),i}$:
   (a) Let $\mathsf{ans} := f(w \parallel u_{t(n),i})$.
   (b) Set $\mathsf{ans}' := \mathsf{ans}$.
   (c) Let $\mathsf{ans}'_1 := 1 - \mathsf{ans}_1$.
   (d) Give ans' as the answer to the query $w \parallel u_{t(n),i}$.
5. Let $\widetilde{\pi}'$ be the output of $\mathcal{P}(1^\lambda, (\mathbbm{x}, \mathbbm{w}), \zeta)$ with the altered answer.
6. Output $(\mathbbm{x}, \widetilde{\pi}')$.

Note that $\widetilde{\mathcal{P}}_{n,w,i}$ has the same query complexity and running time as $\mathcal{P}$.

For the derivation below, we omit the sampling of the verifier's randomness ρ and prover's randomness ζ in the experiment, they are always sampled as follows:

$$\left[\begin{array}{c} \zeta \leftarrow \{0,1\}^{\mathsf{pr}} \\ \rho \leftarrow \{0,1\}^{\mathsf{vr}} \end{array}\right] \ .$$

We can deduce:

$$\sum_{w\in\{0,1\}^{t(n)}} \Pr\left[\begin{array}{l} f \in \mathsf{UF}_{n,w} \\ \wedge \mathcal{V}^{f'}(1^\lambda, 0^n, \pi, \rho) = 1 \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, w) \leftarrow \mathcal{A}^f_{n,w} \\ i \leftarrow [t(n)] \\ f' := \mathsf{Flip}\,(f, w, i) \\ \pi := \mathcal{P}^f((0^n, w), \zeta) \end{array}\right]$$

$$= \sum_{w\in\{0,1\}^{t(n)}} \sum_{f^\star\in\mathsf{UF}_{n,w}} \sum_{i^\star\in[t(n)]} \Pr\left[\begin{array}{l} i = i^\star \\ \wedge f = f^\star \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ i \leftarrow [t(n)] \end{array}\right] \cdot \Pr\left[\mathcal{V}^{f'}(1^\lambda, 0^n, \pi, \rho) = 1 \middle| \begin{array}{l} (0^n, w) \leftarrow \mathcal{A}^{f^\star}_{n,w} \\ f' := \mathsf{Flip}\,(f^\star, w, i^\star) \\ \pi := \mathcal{P}^{f^\star}((0^n, w), \zeta) \end{array}\right]$$

$$=_{[1]} \frac{1}{t(n)} \sum_{w\in\{0,1\}^{t(n)}} \sum_{f^\star\in\mathsf{UF}_{n,w}} \sum_{i^\star\in[t(n)]} \Pr\left[f = f^\star \mid f \leftarrow \mathcal{O}_\lambda\right] \cdot \Pr\left[\mathcal{V}^{f'}(1^\lambda, 0^n, \pi, \rho) = 1 \middle| \begin{array}{l} (0^n, w) \leftarrow \mathcal{A}^{f^\star}_{n,w} \\ f' := \mathsf{Flip}\,(f^\star, w, i^\star) \\ \pi := \mathcal{P}^{f^\star}((0^n, w), \zeta) \end{array}\right]$$

$$= \frac{1}{t(n)} \sum_{w\in\{0,1\}^{t(n)}} \sum_{i^\star\in[t(n)]} \sum_{\substack{f^\star\in\mathsf{UF}_{n,w} \\ f' := \mathsf{Flip}(f^\star, w, i^\star)}} \Pr\left[f = f^\star \mid f \leftarrow \mathcal{O}_\lambda\right] \cdot \Pr\left[\mathcal{V}^{f'}(1^\lambda, 0^n, \pi, \rho) = 1 \middle| \begin{array}{l} (0^n, w) \leftarrow \mathcal{A}^{f^\star}_{n,w} \\ \pi := \mathcal{P}^{f^\star}((0^n, w), \zeta) \end{array}\right]$$

$$= \frac{1}{t(n)} \sum_{w\in\{0,1\}^{t(n)}} \sum_{i^\star\in[t(n)]} \sum_{\substack{f^\star\in\mathsf{UF}_{n,w} \\ f' := \mathsf{Flip}(f^\star, w, i^\star)}} \Pr\left[f = f^\star \mid f \leftarrow \mathcal{O}_\lambda\right] \cdot \Pr\left[\mathcal{V}^{f'}(1^\lambda, 0^n, \pi, \rho) = 1 \middle| \begin{array}{l} (0^n, w) \leftarrow \mathcal{A}^{f^\star}_{n,w} \\ \pi := \mathcal{P}^{f^\star}((0^n, w), \zeta) \\ (0^n, \widetilde{\pi}') := \widetilde{\mathcal{P}}^{f'}_{n,w,i^\star}(\zeta) \end{array}\right]$$

$$=_{[2]} \frac{1}{t(n)} \sum_{w\in\{0,1\}^{t(n)}} \sum_{i^\star\in[t(n)]} \sum_{\substack{f^\star\in\mathsf{UF}_{n,w} \\ f' := \mathsf{Flip}(f^\star, w, i^\star)}} \Pr\left[f = f^\star \mid f \leftarrow \mathcal{O}_\lambda\right] \cdot \Pr\left[\mathcal{V}^{f'}(1^\lambda, 0^n, \widetilde{\pi}', \rho) = 1 \middle| (0^n, \widetilde{\pi}') := \widetilde{\mathcal{P}}^{f'}_{n,w,i^\star}(\zeta)\right]$$

$$\leq_{[3]} \frac{1}{t(n)} \max_{\substack{w\in\{0,1\}^{t(n)} \\ i^\star\in[t(n)]}} \left\{\sum_{f'\in\mathsf{F}_{n,0}} \Pr\left[f = f' \mid f \leftarrow \mathcal{O}_\lambda\right] \cdot \Pr\left[\mathcal{V}^{f'}(1^\lambda, 0^n, \widetilde{\pi}', \rho) = 1 \middle| (0^n, \widetilde{\pi}') := \widetilde{\mathcal{P}}^{f'}_{n,w,i^\star}(\zeta)\right]\right\}$$

$$= \frac{1}{t(n)} \max_{\substack{w\in\{0,1\}^{t(n)} \\ i^\star\in[t(n)]}} \left\{\sum_{f^\star\in\mathsf{F}_{n,0}} \Pr\left[f = f^\star \mid f \leftarrow \mathcal{O}_\lambda\right] \cdot \Pr\left[\mathcal{V}^{f^\star}(1^\lambda, 0^n, \widetilde{\pi}', \rho) = 1 \middle| (0^n, \widetilde{\pi}') := \widetilde{\mathcal{P}}^{f^\star}_{n,w,i^\star}(\zeta)\right]\right\}$$

$$= \frac{1}{t(n)} \max_{\substack{w\in\{0,1\}^{t(n)} \\ i^\star\in[t(n)]}} \left\{\Pr\left[\begin{array}{l} f \in \mathsf{F}_{n,0} \\ \wedge \mathcal{V}^f(1^\lambda, 0^n, \widetilde{\pi}', \rho) = 1 \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, \widetilde{\pi}') := \widetilde{\mathcal{P}}^f_{n,w,i^\star}(\zeta) \end{array}\right]\right\}$$

$$=_{[4]} \frac{1}{t(n)} \max_{\substack{w\in\{0,1\}^{t(n)} \\ i^\star\in[t(n)]}} \left\{\Pr\left[\begin{array}{l} 0^n \notin L(R_f) \\ \wedge \mathcal{V}^f(1^\lambda, 0^n, \widetilde{\pi}', \rho) = 1 \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, \widetilde{\pi}') := \widetilde{\mathcal{P}}^f_{n,w,i^\star}(\zeta) \end{array}\right]\right\}$$

$$\leq_{[5]} \frac{\beta(\lambda, n, \mathsf{pq}, \mathsf{pt})}{t(n)} \quad ,$$

where

- Equality [1] holds because for every $i^\star \in [t(n)]$, $\Pr\left[i = i^\star \mid i \leftarrow [t(n)]\right] = \frac{1}{t(n)}$;
- Equality [2] holds because for every prover randomness $\zeta \in \{0,1\}^{\mathsf{pr}}$, $f \in \mathcal{O}_\lambda$, $w \in \{0,1\}^{t(n)}$, and $i \in [t(n)]$, $(0^n, \mathcal{P}^f((0^n, w), \zeta)) = \widetilde{\mathcal{P}}^{\mathsf{Flip}(f,w,i)}_{n,w,i}(\zeta)$;

- Inequality [3] holds: Let $S := [f' := \mathsf{Flip}(f, w, i) : w \in \{0,1\}^{t(n)}, i \in [t(n)], f \in \mathsf{UF}_{n,w}]$ (S is a multiset). Let $[\mathsf{F}_{n,0}]$ be the multiset casting of $\mathsf{F}_{n,0}$. We know that $S \subseteq [\mathsf{F}_{n,0}]$;
- Equality [4] follows from definition of $\mathsf{F}_{n,0}$;
- Inequality [5] follows from Definition 3.13.

On the other hand,

$$
\begin{aligned}
&\sum_{w\in\{0,1\}^{t(n)}} \Pr\left[\begin{array}{l} f \in \mathsf{UF}_{n,w} \\ \wedge\, \mathcal{V}^{f'}(1^\lambda, 0^n, \pi, \rho) = 1 \end{array} \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, w) \leftarrow \mathcal{A}^f_{n,w} \\ i \leftarrow [t(n)] \\ f' := \mathsf{Flip}(f, w, i) \\ \pi := \mathcal{P}^f((0^n, w), \zeta) \end{array}\right] \\
\geq_{[1]} &\sum_{w\in\{0,1\}^{t(n)}} \left(\begin{array}{l} \Pr\left[\begin{array}{l} \mathcal{V}^{f'}(1^\lambda, 0^n, \pi, \rho) = 1 \\ \text{conditioned on} \\ i \in Q^\star(f, w) \\ \wedge\, f \in \mathsf{UF}_{n,w} \\ \wedge\, \mathcal{V}^{f}(1^\lambda, 0^n, \pi, \rho) = 1 \end{array} \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, w) \leftarrow \mathcal{A}^f_{n,w} \\ i \leftarrow [t(n)] \\ f' := \mathsf{Flip}(f, w, i) \\ \pi := \mathcal{P}^f((0^n, w), \zeta) \end{array}\right] \\ \cdot \Pr\left[\begin{array}{l} i \in Q^\star(f, w) \\ \wedge\, f \in \mathsf{UF}_{n,w} \\ \wedge\, \mathcal{V}^{f}(1^\lambda, 0^n, \pi, \rho) = 1 \end{array} \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, w) \leftarrow \mathcal{A}^f_{n,w} \\ i \leftarrow [t(n)] \\ \pi := \mathcal{P}^f((0^n, w), \zeta) \end{array}\right] \end{array} \right) \\
\geq_{[2]} &\,(1 - p(n)) \cdot \sum_{w\in\{0,1\}^{t(n)}} \Pr\left[\begin{array}{l} i \in Q^\star(f, w) \\ \wedge\, f \in \mathsf{UF}_{n,w} \\ \wedge\, \mathcal{V}^{f}(1^\lambda, 0^n, \pi, \rho) = 1 \end{array} \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, w) \leftarrow \mathcal{A}^f_{n,w} \\ i \leftarrow [t(n)] \\ \pi := \mathcal{P}^f((0^n, w), \zeta) \end{array}\right] \\
= &\,(1 - p(n)) \cdot \sum_{\substack{w\in\{0,1\}^{t(n)} \\ f^\star \in \mathsf{UF}_{n,w} \\ i^\star \in [t(n)] \\ i^\star \in Q^\star(f,w)}} \Pr\left[\begin{array}{l} i = i^\star \\ \wedge\, f = f^\star \end{array} \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ i \leftarrow [t(n)] \end{array}\right] \cdot \Pr\left[\mathcal{V}^{f^\star}(1^\lambda, 0^n, \pi, \rho) = 1 \;\middle|\; \begin{array}{l} (0^n, w) \leftarrow \mathcal{A}^{f^\star}_{n,w} \\ \pi := \mathcal{P}^{f^\star}((0^n, w), \zeta) \end{array}\right] \\
=_{[3]} &\,(1 - p(n)) \cdot \frac{1}{t(n)} \cdot \sum_{\substack{w\in\{0,1\}^{t(n)} \\ f^\star \in \mathsf{UF}_{n,w} \\ i^\star \in [t(n)] \\ i^\star \in Q^\star(f,w)}} \Pr\left[f = f^\star \mid f \leftarrow \mathcal{O}_\lambda\right] \cdot \Pr\left[\mathcal{V}^{f^\star}(1^\lambda, 0^n, \pi, \rho) = 1 \;\middle|\; \begin{array}{l} (0^n, w) \leftarrow \mathcal{A}^{f^\star}_{n,w} \\ \pi := \mathcal{P}^{f^\star}((0^n, w), \zeta) \end{array}\right] \\
\geq_{[4]} &\,(1 - p(n)) \cdot \frac{t(n) - \frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)} \cdot \sum_{\substack{w\in\{0,1\}^{t(n)} \\ f^\star \in \mathsf{UF}_{n,w}}} \Pr\left[f = f^\star \mid f \leftarrow \mathcal{O}_\lambda\right] \cdot \Pr\left[\mathcal{V}^{f^\star}(1^\lambda, 0^n, \pi, \rho) = 1 \;\middle|\; \begin{array}{l} (0^n, w) \leftarrow \mathcal{A}^{f^\star}_{n,w} \\ \pi := \mathcal{P}^{f^\star}((0^n, w), \zeta) \end{array}\right] \\
= &\,(1 - p(n)) \cdot \frac{t(n) - \frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)} \cdot \sum_{w\in\{0,1\}^{t(n)}} \Pr\left[\begin{array}{l} f \in \mathsf{UF}_{n,w} \\ \wedge\, \mathcal{V}^{f}(1^\lambda, 0^n, \pi, \rho) = 1 \end{array} \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, w) \leftarrow \mathcal{A}^f_{n,w} \\ \pi := \mathcal{P}^f((0^n, w), \zeta) \end{array}\right] ,
\end{aligned}
$$

where

- Inequality [1] follows from definition of conditional probability;
- Inequality [2] follows from Eq. (8);
- Equality [3] holds because every $i^\star \in [t(n)]$, $\Pr[i = i^\star \mid i \leftarrow [t(n)]] = \frac{1}{t(n)}$;
- Inequality [4] follows from Eq. (7).

Moreover,

$$
\begin{aligned}
&\sum_{w \in \{0,1\}^{t(n)}} \Pr\left[\begin{array}{l} f \in \mathsf{UF}_{n,w} \\ \wedge \mathcal{V}^f(1^\lambda, 0^n, \pi, \rho) = 1 \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, w) \leftarrow \mathcal{A}^f_{n,w} \\ \pi := \mathcal{P}^f((0^n, w), \zeta) \end{array}\right] \\
&=_{[1]} \Pr\left[\begin{array}{l} f \in \mathsf{UF}_{n} \\ \wedge \mathcal{V}^f(1^\lambda, 0^n, \pi, \rho) = 1 \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, w) \leftarrow \mathcal{A}^f_{n} \\ \pi := \mathcal{P}^f((0^n, w), \zeta) \end{array}\right] \\
&=_{[2]} \Pr\left[\begin{array}{l} f \in \mathsf{F}_{n,1} \\ \wedge \mathcal{V}^f(1^\lambda, 0^n, \pi, \rho) = 1 \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, w) \leftarrow \mathcal{A}^f_{n} \\ \pi := \mathcal{P}^f((0^n, w), \zeta) \end{array}\right] - \Pr\left[\begin{array}{l} f \in \mathsf{F}_{n,1} \setminus \mathsf{UF}_n \\ \wedge \mathcal{V}^f(1^\lambda, 0^n, \pi, \rho) = 1 \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, w) \leftarrow \mathcal{A}^f_{n} \\ \pi := \mathcal{P}^f((0^n, w), \zeta) \end{array}\right] \\
&\geq \Pr\left[\begin{array}{l} f \in \mathsf{F}_{n,1} \\ \wedge \mathcal{V}^f(1^\lambda, 0^n, \pi, \rho) = 1 \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, w) \leftarrow \mathcal{A}^f_{n} \\ \pi := \mathcal{P}^f((0^n, w), \zeta) \end{array}\right] - \Pr[f \in \mathsf{F}_{n,1} \setminus \mathsf{UF}_n \mid f \leftarrow \mathcal{O}_\lambda] \\
&=_{[3]} \Pr[f \in \mathsf{F}_{n,1} \mid f \leftarrow \mathcal{O}_\lambda] - \Pr\left[\begin{array}{l} f \in \mathsf{F}_{n,1} \\ \wedge \mathcal{V}^f(1^\lambda, 0^n, \pi, \rho) = 0 \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, w) \leftarrow \mathcal{A}^f_{n} \\ \pi := \mathcal{P}^f((0^n, w), \zeta) \end{array}\right] - \Pr[f \in \mathsf{F}_{n,1} \setminus \mathsf{UF}_n \mid f \leftarrow \mathcal{O}_\lambda] \\
&= \Pr[f \in \mathsf{UF}_{n} \mid f \leftarrow \mathcal{O}_\lambda] - \Pr\left[\begin{array}{l} f \in \mathsf{F}_{n,1} \\ \wedge \mathcal{V}^f(1^\lambda, 0^n, \pi, \rho) = 0 \end{array} \middle| \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, w) \leftarrow \mathcal{A}^f_{n} \\ \pi := \mathcal{P}^f((0^n, w), \zeta) \end{array}\right] \\
&\geq_{[4]} \Pr[f \in \mathsf{UF}_{n} \mid f \leftarrow \mathcal{O}_\lambda] - \alpha(\lambda, n) \\
&=_{[5]} \left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}-1} - \alpha(\lambda, n) \quad ,
\end{aligned}
$$

where

- Equality [1] holds because for every $w, w' \in \{0,1\}^{t(n)}$ where $w \neq w'$, $\mathsf{UF}_{n,w} \cap \mathsf{UF}_{n,w'} = \emptyset$. Moreover, $\cup_{w \in \{0,1\}^{t(n)}} \mathsf{UF}_{n,w} = \mathsf{UF}_n$;
- Equality [2] follows from the law of total probability;
- Equality [3] follows from the law of total probability;
- Inequality [4] follows from Definition 3.12;
- Equality [5] holds because $\mu_{\mathcal{O}_\lambda}(\mathsf{UF}_n) = \left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}-1} \cdot \frac{1}{2^{t(n)}} \cdot 2^{t(n)} = \left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}-1}$.

We can conclude that

$$(1 - p(n)) \cdot \frac{t(n) - \frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)} \cdot \left(\left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}-1} - \alpha(\lambda, n) \right) \leq \frac{\beta(\lambda, n, \mathsf{pq}, \mathsf{pt})}{t(n)} \ ,$$

a contradiction.

7 The case of IARG in the ROM

We discuss how to generalize results in Sections 4 to 6 to interactive arguments. The proofs follow the same ideas with slight twist to work for interactive setting.

7.1 Relativized interactive arguments in the ROM

Definition 7.1. *A* **relativized interactive argument** *relative to the random oracle $\mathcal{O}$ for an relativized relation $R_{\mathcal{O}}$ is a tuple of algorithms $\mathsf{IARG} = (\mathcal{P}, \mathcal{V})$ that works as follows: For every $\lambda \in \mathbb{N}$ and $f \in \mathcal{O}_\lambda$, $\mathcal{P}^f(1^\lambda, \mathbb{x}, \mathbb{w})$ and $\mathcal{V}^f(1^\lambda, \mathbb{x})$ interact with each other, at the end of the interaction, $\mathcal{V}$ outputs a decision* $\mathsf{d} \in \{0, 1\}$.

Definition 7.2 (Completeness)**.** *For every security parameter $\lambda \in \mathbb{N}$, instance size bound $n \in \mathbb{N}$, and adversary $\mathcal{A}$,*

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

Definition 7.3 (Soundness)**.** *For every security parameter $\lambda \in \mathbb{N}$, instance size bound $n \in \mathbb{N}$, 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 $\widetilde{\mathcal{P}}$,*

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

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

- the *round complexity* k is the maximum number of rounds the prover $\mathcal{P}$ and the verifier $\mathcal{V}$ is allowed to interact;
- the *prover communication* $\mathsf{pc}(\lambda, n)$ is the maximum number of bits the prover $\mathcal{P}$ is allowed to send to the verifier $\mathcal{V}$;
- 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 7.4 (Relativized IARG in the ROM)**.** *Let $\mathcal{O}$ be the random oracle. The complexity class*

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

is the set of relativized relations $R_{\mathcal{O}} = \{R_f\}_{\lambda\in\mathbb{N},f\in\mathcal{O}_\lambda}$ that admits 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.*

7.2 Separation of DTIME and IARG

Theorem 7.5. *Let $\mathcal{O}$ be the random oracle. For every time-constructible function $t\colon \mathbb{N}\to\mathbb{N}$, argument verifier query bound $\mathsf{vq}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument verifier time bound $\mathsf{vt}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument prover communication complexity $\mathsf{pc} : \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument honest prover query bound $\mathsf{pq}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument honest prover time bound $\mathsf{pt}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument completeness $\alpha\colon \mathbb{N}\times\mathbb{N}\to[0,1]$ and argument soundness $\beta\colon \mathbb{N}\times\mathbb{N}\times\mathbb{N}\times\mathbb{N}\to[0,1]$,*

$$\mathsf{DTIME}^{\mathcal{O}}\left[t(n)\right] \not\subseteq \mathsf{IARG}^{\mathcal{O}}\left[\begin{array}{lcll} \text{completeness error} & \alpha & = & \alpha(\lambda,n) \\ \text{soundness error} & \beta & = & \beta(\lambda,n,\mathsf{q}_{\widetilde{\mathcal{P}}},\mathsf{t}_{\widetilde{\mathcal{P}}}) \\ \text{verifier query bound} & \mathsf{vq} & = & \mathsf{vq}(\lambda,n) \\ \text{verifier running time} & \mathsf{vt} & = & \mathsf{vt}(\lambda,n) \\ \text{prover communication} & \mathsf{pc} & = & \mathsf{pc}(\lambda,n) \\ \text{honest prover query bound} & \mathsf{pq} & = & \mathsf{pq}(\lambda,n) \\ \text{honest prover time bound} & \mathsf{pt} & = & \mathsf{pt}(\lambda,n) \end{array}\right],$$

where there exists $\lambda\in\mathbb{N}$, $n\in\mathbb{N}$ and some function $p\colon \mathbb{N}\to(0,1]$ such that the following holds:

- $0\le\alpha(\lambda,n)\le 1$,
- $0\le\beta(\lambda,n,t(n)+\mathsf{pq},t(n)+\mathsf{pt}) < (1-\alpha(\lambda,n))\cdot(1-p(n))\cdot\frac{t(n)-\frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)}$*, and*
- $(1-\alpha(\lambda,n))\cdot(1-p(n))\cdot\frac{t(n)-\frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)}\le 1$.

The corollary below follows by setting $p(n)<\frac{1}{2}$ for all $n\in\mathbb{N}$ in Theorem 6.1.

Corollary 7.6. *Let $\mathcal{O}$ be the random oracle. For every time-constructible function $t\colon \mathbb{N} \times \mathbb{N}$, $\lambda \in \mathbb{N}$ and $n \in \mathbb{N}$,*

$$\mathsf{DTIME}^{\mathcal{O}}[t(n)] \not\subseteq \mathsf{IARG}^{\mathcal{O}} \left[\begin{array}{lll} \text{completeness error} & \alpha & = \; 1/3 \\ \text{soundness error} & \beta & = \; \begin{cases} 1/3 & \mathsf{q}_{\widetilde{\mathcal{P}}} = \mathsf{poly}(\lambda, t(n)), \mathsf{t}_{\widetilde{\mathcal{P}}} = \mathsf{poly}(\lambda, t(n)) \\ 1 & \textit{otherwise} \end{cases} \\ \text{verifier query bound} & \mathsf{vq} & = \; o(t(n)) \\ \text{verifier running time} & \mathsf{vt} & \\ \text{prover communication} & \mathsf{pc} & \\ \text{honest prover query bound} & \mathsf{pq} & = \; \mathsf{poly}(\lambda, t(n)) \\ \text{honest prover time bound} & \mathsf{pt} & = \; \mathsf{poly}(\lambda, t(n)) \end{array}\right] .$$

To prove Theorem 7.5, it suffices to prove the following lemma:

Lemma 7.7. *Let $\mathcal{O}$ be the random oracle. Let $L_{\mathcal{O}}$ be defined as in Definition 4.3. Fix time-constructible function $t\colon \mathbb{N} \to \mathbb{N}$, argument verifier query bound $\mathsf{vq}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument verifier time bound $\mathsf{vt}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument prover communication complexity $\mathsf{pc}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument honest prover query bound $\mathsf{pq}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument honest prover time bound $\mathsf{pt}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument completeness $\alpha\colon \mathbb{N} \times \mathbb{N} \to [0,1]$ and argument soundness $\beta\colon \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to [0,1]$.*

There is no relativized interactive argument $(\mathcal{P}, \mathcal{V})$ for the relativized language $L_{\mathcal{O}}$ with completeness error α, soundness error β, argument verifier running time vt, argument prover communication complexity pc, argument honest prover query bound pq, and argument honest prover time bound pt if there exists $\lambda \in \mathbb{N}$, $n \in \mathbb{N}$ and $p\colon \mathbb{N} \to [0,1]$ such that the following holds:

- $0 \leq \alpha(\lambda, n) \leq 1$,
- $0 \leq \beta(\lambda, n, t(n) + \mathsf{pq}, t(n) + \mathsf{pt}) < (1 - \alpha(\lambda, n)) \cdot (1 - p(n)) \cdot \frac{t(n) - \frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)}$, *and*
- $(1 - \alpha(\lambda, n)) \cdot (1 - p(n)) \cdot \frac{t(n) - \frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)} \leq 1$.

Proof of Lemma 7.7. We adapt the proof of Lemma 4.4.

Consider the set of queries Q:

$$Q := \{q(i,j)\}_{i \in [n/2], j \in [t^*(n)]} .$$

Note that $|Q| = t(n)$. For every $f \in \mathcal{O}_\lambda$ and $y \in \{0,1\}^{n/2}$, define the set $Q^\star(f,y)$ as follows:

$$Q^\star(f,y) := \left\{ q \in Q : \Pr\left[q \in \mathsf{tr} \;\middle|\; \begin{array}{l} \rho \leftarrow \{0,1\}^{\mathsf{vr}} \\ \zeta \leftarrow \{0,1\}^{\mathsf{pr}} \\ \text{For } i \in [\mathsf{k}] : \\ \quad a_i \xleftarrow{\mathsf{tr}_i} \mathcal{V}^f(1^\lambda, (0^{n/2}, y), \rho, (b_j)_{j<i}) \\ \quad b_i := \mathcal{P}^f(1^\lambda, (0^{n/2}, y), \zeta, (a_j)_{j \leq i}) \\ \mathsf{d} \xleftarrow{\mathsf{tr}_\mathsf{k}} \mathcal{V}^f(1^\lambda, (0^{n/2}, y), \rho, (b_i)_{i \in [\mathsf{k}]}) \\ \mathsf{tr} := \mathsf{tr}_1 \,\|\, \cdots \,\|\, \mathsf{tr}_\mathsf{k} \end{array} \right] < p(n) \right\} .$$

We can show that

$$|Q^\star(f,y)| \geq t(n) - \frac{\mathsf{vq}(\lambda, n)}{p(n)} . \tag{9}$$

For every $f \in \mathcal{O}_\lambda$, $i \in [n/2]$ and $j \in [t^*(n)]$, we define $\mathsf{Flip}(f,i,j)$ as in the proof of Lemma 4.5. We define $\widetilde{\mathcal{P}}_{n,i,j}$ as follows:

$\widetilde{\mathcal{P}}^{f}_{n,i,j}$:

1. Set $x := 0^{n/2}$.
2. Every time a query is made to f at $x \parallel u_{t(n),(i-1)\cdot t^*(n)+j}$, use the answer of $\mathsf{Flip}(f,i,j)$ instead.
3. Compute $y := F_{f,n}(x)$.
4. Set $\mathbb{x} := (x,y)$.
5. Simulate the honest prover $\mathcal{P}(1^\lambda,(x,y))$ with the altered answer and interact with $\mathcal{V}$.

Then, we can deduce from Eq. (9) and Definitions 7.2 and 7.3 that

$$
\begin{aligned}
&(1-\alpha(\lambda,n))\cdot\frac{t(n)-\frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)}\cdot(1-p(n))\\
&\leq \Pr\left[\begin{array}{c|l}
 & f \leftarrow \mathcal{O}_\lambda \\
 & (i,j) \leftarrow [n/2]\times[t^*(n)] \\
 & \rho \leftarrow \{0,1\}^{\mathsf{vr}} \\
(0^{n/2},y') \notin L_{f'} & \zeta \leftarrow \{0,1\}^{\mathsf{pr}} \\
\wedge\, \mathsf{d}' = 1 & f' := \mathsf{Flip}(f,i,j) \\
 & ((0^{n/2},y'),\mathsf{aux}) := \widetilde{\mathcal{P}}^{f'}_{n,i,j}(\zeta) \\
 & \mathsf{d}' := \langle \widetilde{\mathcal{P}}^{f'}_{n,i,j}(\mathsf{aux}), \mathcal{V}^{f'}(1^\lambda,(0^{n/2},y'),\rho)\rangle
\end{array}\right]\\
&\leq \beta(\lambda,n,t(n)+\mathsf{pq},t(n)+\mathsf{pt})\ ,
\end{aligned}
$$

a contradiction. □

7.3 Separation of NTIME and IARG

Theorem 7.8. *Let $\mathcal{O}$ be the random oracle. For every time-constructible function $t\colon \mathbb{N}\to\mathbb{N}$, argument verifier query bound $\mathsf{vq}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument verifier time bound $\mathsf{vt}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument prover communication complexity $\mathsf{pc}:\mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument honest prover query bound $\mathsf{pq}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument honest prover time bound $\mathsf{pt}\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, argument completeness $\alpha\colon \mathbb{N}\times\mathbb{N}\to[0,1]$ and argument soundness $\beta\colon \mathbb{N}\times\mathbb{N}\times\mathbb{N}\times\mathbb{N}\to[0,1]$,*

$$
\mathsf{NTIME}^{\mathcal{O}}[t(n)] \not\subseteq \mathsf{IARG}^{\mathcal{O}}\left[\begin{array}{lcl}
\text{completeness error} & \alpha & = \alpha(\lambda,n) \\
\text{soundness error} & \beta & = \beta(\lambda,n,\mathsf{q}_{\widetilde{\mathcal{P}}},\mathsf{t}_{\widetilde{\mathcal{P}}}) \\
\text{verifier query bound} & \mathsf{vq} & = \mathsf{vq}(\lambda,n) \\
\text{verifier running time} & \mathsf{vt} & = \mathsf{vt}(\lambda,n) \\
\text{prover communication} & \mathsf{pc} & = \mathsf{pc}(\lambda,n) \\
\text{honest prover query bound} & \mathsf{pq} & = \mathsf{pq}(\lambda,n) \\
\text{honest prover time bound} & \mathsf{pt} & = \mathsf{pt}(\lambda,n)
\end{array}\right],
$$

where there exists $\lambda\in\mathbb{N}$, $n\in\mathbb{N}$ and some function $p\colon \mathbb{N}\to(0,1]$ such that the following holds:

- $0 \leq \beta(\lambda,n,\mathsf{pq},\mathsf{pt}) < \left(1-\frac{1}{2^{t(n)}}\right)^{2^{t(n)}}$,
- $0 \leq \frac{2^{t(n)}-1}{2^{t(n)}}\cdot\alpha(\lambda,n) < (1-t(n)\cdot p(n))\cdot\frac{2^{t(n)}-\frac{2^{O(\mathsf{pc}(\lambda,n))}\cdot\mathsf{vq}(\lambda,n)}{p(n)}}{2^{t(n)}}\cdot\left(\left(1-\frac{1}{2^{t(n)}}\right)^{2^{t(n)}}-\beta(\lambda,n,\mathsf{pq},\mathsf{pt})\right)$, *and*

- $(1 - t(n) \cdot p(n)) \cdot \frac{2^{t(n)} - \frac{2^{O(\mathsf{pc}(\lambda,n))} \cdot \mathsf{vq}(\lambda,n)}{p(n)}}{2^{t(n)}} \cdot \left(\left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}} - \beta(\lambda, n, \mathsf{pq}, \mathsf{pt})\right) \leq 1.$

The corollary below follows by setting $p(n) < \frac{1}{2t(n)}$ for all $n \in \mathbb{N}$ in Theorem 7.8.

Corollary 7.9. *Let $\mathcal{O}$ be the random oracle. For every time-constructible function $t\colon \mathbb{N} \times \mathbb{N}$, $\lambda \in \mathbb{N}$ and $n \in \mathbb{N}$,*

$$\mathsf{NTIME}^{\mathcal{O}}\left[t(n)\right] \not\subseteq \mathsf{IARG}^{\mathcal{O}} \left[\begin{array}{lll} \text{completeness error} & \alpha & = & 1/3 \\ \text{soundness error} & \beta & = & \begin{cases} 1/3 & \mathsf{q}_{\widetilde{\mathcal{P}}} = \mathsf{poly}(\lambda, t(n)), \mathsf{t}_{\widetilde{\mathcal{P}}} = \mathsf{poly}(\lambda, t(n)) \\ 1 & \textit{otherwise} \end{cases} \\ \text{verifier query bound} & \mathsf{vq} & = & \mathsf{poly}(\lambda, t(n)) \\ \text{verifier running time} & \mathsf{vt} & & \\ \text{prover communication} & \mathsf{pc} & = & o(t(n)) \\ \text{honest prover query bound} & \mathsf{pq} & = & \mathsf{poly}(\lambda, t(n)) \\ \text{honest prover time bound} & \mathsf{pt} & = & \mathsf{poly}(\lambda, t(n)) \end{array}\right].$$

To prove Theorem 7.8, it suffices to show the following lemma:

Lemma 7.10. *Let $\mathcal{O}$ be the random oracle. Let $R_{\mathcal{O}}$ be defined as in Definition 5.3. Fix time-constructible function $t\colon \mathbb{N} \to \mathbb{N}$, argument verifier query bound $\mathsf{vq}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument verifier time bound $\mathsf{vt}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument prover communication complexity $\mathsf{pc}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument honest prover query bound $\mathsf{pq}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument honest prover time bound $\mathsf{pt}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument completeness $\alpha\colon \mathbb{N} \times \mathbb{N} \to [0,1]$ and argument soundness $\beta\colon \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to [0,1]$.*

There is no relativized interactive argument $(\mathcal{P}, \mathcal{V})$ for the relativized relation $R_{\mathcal{O}}$ with completeness error α, soundness error β, argument verifier running time vt, argument prover communication complexity pc, argument honest prover query bound pq, and argument honest prover time bound pt if there exists $\lambda \in \mathbb{N}$, $n \in \mathbb{N}$ and $p\colon \mathbb{N} \to [0,1]$ such that the following holds:

- $0 \leq \beta(\lambda, n, \mathsf{pq}, \mathsf{pt}) < \left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}}$,
- $0 \leq \frac{2^{t(n)}-1}{2^{t(n)}} \cdot \alpha(\lambda, n) < (1 - t(n) \cdot p(n)) \cdot \frac{2^{t(n)} - \frac{2^{O(\mathsf{pc}(\lambda,n))} \cdot \mathsf{vq}(\lambda,n)}{p(n)}}{2^{t(n)}} \cdot \left(\left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}} - \beta(\lambda, n, \mathsf{pq}, \mathsf{pt})\right)$, *and*
- $(1 - t(n) \cdot p(n)) \cdot \frac{2^{t(n)} - \frac{2^{O(\mathsf{pc}(\lambda,n))} \cdot \mathsf{vq}(\lambda,n)}{p(n)}}{2^{t(n)}} \cdot \left(\left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}} - \beta(\lambda, n, \mathsf{pq}, \mathsf{pt})\right) \leq 1.$

Proof of Lemma 7.10. The proof follows the same idea as the proof of Lemma 5.5.

For interactive arguments, we measure the prover communication complexity instead of the argument size. In particular, for every $f \in \mathcal{O}_\lambda$, we can define the set $Q^\star(f)$ the set of $w \in \{0,1\}^{t(n)}$ that satisfies the following:

$$\forall i \in [t(n)], \quad \sum_{(b_1,\ldots,b_{\mathsf{k}}) \in \{0,1\}^{\leq \mathsf{pc}}} \Pr\left[w \parallel u_{t(n),i} \in \mathsf{tr} \;\middle|\; \begin{array}{l} \rho \leftarrow \{0,1\}^{\mathsf{vr}} \\ \text{For } i \in [\mathsf{k}]: \\ \quad a_i \xleftarrow{\mathsf{tr}_i} \mathcal{V}^f(1^\lambda, 0^n, \rho, (b_j)_{j<i}) \\ \mathsf{d} \xleftarrow{\mathsf{tr}_\mathsf{k}} \mathcal{V}^f(1^\lambda, 0^n, \rho, (b_i)_{i \in [\mathsf{k}]}) \\ \mathsf{tr} := \mathsf{tr}_1 \parallel \cdots \parallel \mathsf{tr}_\mathsf{k} \end{array} \right] < p(n) \;\; ,$$

where $(\cdot) \xleftarrow{\mathsf{tr}} A^f(\cdot)$ means tr is the set of all queries made by algorithm A to the oracle f.

Similarly, we can define the set $Q_c(f)$ as the set of $w \parallel u \in \{0,1\}^{t(n)+\lceil \log t(n) \rceil}$ such that

$$\sum_{(b_1,\ldots,b_{\mathsf{k}})\in\{0,1\}^{\leq \mathsf{pc}}} \Pr\left[w \parallel u \in \mathsf{tr} \;\middle|\; \begin{array}{l} \rho \leftarrow \{0,1\}^{\mathsf{vr}} \\ \text{For } i \in [\mathsf{k}] : \\ \quad a_i \xleftarrow{\mathsf{tr}_i} \mathcal{V}^f(1^\lambda, 0^n, \rho, (b_j)_{j<i}) \\ \mathsf{d} \xleftarrow{\mathsf{tr}_{\mathsf{k}}} \mathcal{V}^f(1^\lambda, 0^n, \rho, (b_i)_{i\in[\mathsf{k}]}) \\ \mathsf{tr} := \mathsf{tr}_1 \parallel \cdots \parallel \mathsf{tr}_{\mathsf{k}} \end{array} \right] \geq p(n) \ .$$

Since $\mathcal{V}^f$ can make at most $\mathsf{vq}(\lambda, n)$ queries to f:

$$\sum_{w\parallel u\in\{0,1\}^{t(n)+\lceil \log t(n) \rceil}} \sum_{(b_1,\ldots,b_{\mathsf{k}})\in\{0,1\}^{\leq \mathsf{pc}}} \Pr\left[w \parallel u \in \mathsf{tr} \;\middle|\; \begin{array}{l} \rho \leftarrow \{0,1\}^{\mathsf{vr}} \\ \text{For } i \in [\mathsf{k}] : \\ \quad a_i \xleftarrow{\mathsf{tr}_i} \mathcal{V}^f(1^\lambda, 0^n, \rho, (b_j)_{j<i}) \\ \mathsf{d} \xleftarrow{\mathsf{tr}_{\mathsf{k}}} \mathcal{V}^f(1^\lambda, 0^n, \rho, (b_i)_{i\in[\mathsf{k}]}) \\ \mathsf{tr} := \mathsf{tr}_1 \parallel \cdots \parallel \mathsf{tr}_{\mathsf{k}} \end{array} \right] \leq 2^{O(\mathsf{pc})} \cdot \mathsf{vq}(\lambda, n) \ .$$

Hence,

$$|Q_c(f)| \leq \frac{2^{O(\mathsf{pc})} \cdot \mathsf{vq}(\lambda, n)}{p(n)} \ ,$$

which implies that

$$|Q^\star(f)| \geq 2^{t(n)} - |Q_c(f)| \geq 2^{t(n)} - \frac{2^{O(\mathsf{pc})} \cdot \mathsf{vq}(\lambda, n)}{p(n)} \ . \tag{10}$$

Moreover, we can deduce from definition of $Q^\star(f)$ that, for every $(b_1, \ldots, b_{\mathsf{k}}) \in \{0,1\}^{\leq \mathsf{pc}}$,

$$\Pr\left[\forall i \in [t(n)], w \parallel u_{t(n),i} \notin \mathsf{tr} \;\middle|\; \begin{array}{l} \rho \leftarrow \{0,1\}^{\mathsf{vr}} \\ \text{For } i \in [\mathsf{k}] : \\ \quad a_i \xleftarrow{\mathsf{tr}_i} \mathcal{V}^f(1^\lambda, 0^n, \rho, (b_j)_{j<i}) \\ \mathsf{d} \xleftarrow{\mathsf{tr}_{\mathsf{k}}} \mathcal{V}^f(1^\lambda, 0^n, \rho, (b_i)_{i\in[\mathsf{k}]}) \\ \mathsf{tr} := \mathsf{tr}_1 \parallel \cdots \parallel \mathsf{tr}_{\mathsf{k}} \end{array} \right] \geq 1 - t(n) \cdot p(n) \ . \tag{11}$$

For every $f \in \mathcal{O}_\lambda$ and $w \in \{0,1\}^{t(n)}$, we define $\mathcal{A}_{n,w}$ as follows:

$\mathcal{A}_{n,w}^f$: Output $(0^n, w)$.

We define $\mathsf{SetZero}(f, w)$ as in the proof of Lemma 5.5.

We can deduce from Eqs. (10) and (11) and Definitions 7.2 and 7.3 as in the proof of Lemma 5.5 that

$$\begin{aligned} &(1 - t(n) \cdot p(n)) \cdot \frac{2^{t(n)} - \frac{2^{O(\mathsf{pc}(\lambda,n))} \cdot \mathsf{vq}(\lambda,n)}{p(n)}}{2^{t(n)}} \cdot \left(\left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}} - \beta(\lambda, n, \mathsf{pq}, \mathsf{pt}) \right) \\ &\leq \Pr\left[\begin{array}{l} 0^n \notin L(R_f) \\ \wedge\, \mathsf{d}' = 0 \end{array} \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ w \leftarrow \{0,1\}^{t(n)} \\ f' := \mathsf{SetZero}(f, w) \\ (0^n, w) := \mathcal{A}_{n,w}^f \\ \rho \leftarrow \{0,1\}^{\mathsf{vr}} \\ \zeta \leftarrow \{0,1\}^{\mathsf{pr}} \\ \mathsf{d}' := \langle \mathcal{P}^{f'}(1^\lambda, (0^n, w), \zeta), \mathcal{V}^{f'}(1^\lambda, 0^n, \rho) \rangle \end{array} \right] \end{aligned}$$

$$\leq \frac{2^{t(n)}-1}{2^{t(n)}} \cdot \alpha(\lambda, n) \ ,$$

□

a contradiction.

7.4 Alternative separation of NTIME and IARG

Theorem 7.11. *Let $\mathcal{O}$ be the random oracle. For every time-constructible function $t\colon \mathbb{N} \to \mathbb{N}$, argument verifier query bound $\mathsf{vq}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument verifier time bound $\mathsf{vt}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument prover communication complexity $\mathsf{pc} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument honest prover query bound $\mathsf{pq}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument honest prover time bound $\mathsf{pt}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument completeness $\alpha\colon \mathbb{N} \times \mathbb{N} \to [0,1]$ and argument soundness $\beta\colon \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to [0,1]$,*

$$\mathsf{NTIME}^{\mathcal{O}}[t(n)] \not\subseteq \mathsf{IARG}^{\mathcal{O}} \begin{bmatrix} \text{completeness error} & \alpha & = & \alpha(\lambda, n) \\ \text{soundness error} & \beta & = & \beta(\lambda, n, \mathsf{q}_{\widetilde{\mathcal{P}}}, \mathsf{t}_{\widetilde{\mathcal{P}}}) \\ \text{verifier query bound} & \mathsf{vq} & = & \mathsf{vq}(\lambda, n) \\ \text{verifier running time} & \mathsf{vt} & = & \mathsf{vt}(\lambda, n) \\ \text{prover communication} & \mathsf{pc} & = & \mathsf{pc}(\lambda, n) \\ \text{honest prover query bound} & \mathsf{pq} & = & \mathsf{pq}(\lambda, n) \\ \text{honest prover time bound} & \mathsf{pt} & = & \mathsf{pt}(\lambda, n) \end{bmatrix} ,$$

where there exists $\lambda \in \mathbb{N}$, $n \in \mathbb{N}$ and some function $p\colon \mathbb{N} \to (0,1]$ such that the following holds:

- $0 \leq \alpha(\lambda, n) < \left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}-1}$,
- $0 \leq \beta(\lambda, n, \mathsf{pq}, \mathsf{pt}) < (1 - p(n)) \cdot (t(n) - \frac{\mathsf{vq}(\lambda,n)}{p(n)}) \cdot \left(\left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}-1} - \alpha(\lambda, n)\right)$,
- $(1 - p(n)) \cdot \frac{t(n) - \frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)} \cdot \left(\left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}-1} - \alpha(\lambda, n)\right) \leq 1$.

The corollary below follows by setting $p(n) < \frac{1}{2}$ for all $n \in \mathbb{N}$ in Theorem 6.1.

Corollary 7.12. *Let $\mathcal{O}$ be the random oracle. For every time-constructible function $t\colon \mathbb{N} \times \mathbb{N}$, $\lambda \in \mathbb{N}$ and $n \in \mathbb{N}$,*

$$\mathsf{NTIME}^{\mathcal{O}}[t(n)] \not\subseteq \mathsf{IARG}^{\mathcal{O}} \begin{bmatrix} \text{completeness error} & \alpha & = & 1/3 \\ \text{soundness error} & \beta & = & \begin{cases} 1/3 & \mathsf{q}_{\widetilde{\mathcal{P}}} = \mathsf{poly}(\lambda, t(n)), \mathsf{t}_{\widetilde{\mathcal{P}}} = \mathsf{poly}(\lambda, t(n)) \\ 1 & \textit{otherwise} \end{cases} \\ \text{verifier query bound} & \mathsf{vq} & = & o(t(n)) \\ \text{verifier running time} & \mathsf{vt} & & \\ \text{prover communication} & \mathsf{pc} & & \\ \text{honest prover query bound} & \mathsf{pq} & = & \mathsf{poly}(\lambda, t(n)) \\ \text{honest prover time bound} & \mathsf{pt} & = & \mathsf{poly}(\lambda, t(n)) \end{bmatrix} .$$

To prove Theorem 7.11, it suffices to prove the following lemma:

Lemma 7.13. *Let $\mathcal{O}$ be the random oracle. Let $R_{\mathcal{O}}$ be defined as in Definition 5.3. Fix time-constructible function $t\colon \mathbb{N} \to \mathbb{N}$, argument verifier query bound $\mathsf{vq}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument verifier time bound $\mathsf{vt}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument prover communication complexity $\mathsf{pc}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument honest prover query bound $\mathsf{pq}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument honest prover time bound $\mathsf{pt}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, argument completeness $\alpha\colon \mathbb{N} \times \mathbb{N} \to [0,1]$ and argument soundness $\beta\colon \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to [0,1]$.*

There is no relativized interactive argument $(\mathcal{P}, \mathcal{V})$ for the relativized relation $R_{\mathcal{O}}$ with completeness error α, soundness error β, argument verifier running time vt, argument prover communication complexity pc, argument honest prover query bound pq, and argument honest prover time bound pt if there exists $\lambda \in \mathbb{N}$, $n \in \mathbb{N}$ and $p\colon \mathbb{N} \to [0,1]$ such that the following holds:

- $0 \leq \alpha(\lambda, n) < \left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}-1}$,
- $0 \leq \beta(\lambda, n, \mathsf{pq}, \mathsf{pt}) < (1 - p(n)) \cdot (t(n) - \frac{\mathsf{vq}(\lambda,n)}{p(n)}) \cdot \left(\left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}-1} - \alpha(\lambda, n)\right)$,
- $(1 - p(n)) \cdot \frac{t(n) - \frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)} \cdot \left(\left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}-1} - \alpha(\lambda, n)\right) \leq 1$.

Proof of Lemma 7.13. We adapt the proof of Lemma 6.3.

For every $f \in \mathcal{O}_\lambda$ and $w \in \{0,1\}^{t(n)}$, we can define the set $Q^\star(f, w)$ as the set of $i \in [t(n)]$ such that

$$\Pr\left[w \parallel u_{t(n),i} \in \mathsf{tr} \;\middle|\; \begin{array}{l} \rho \leftarrow \{0,1\}^{\mathsf{vr}} \\ \zeta \leftarrow \{0,1\}^{\mathsf{pr}} \\ \text{For } i \in [\mathsf{k}]: \\ \quad a_i \xleftarrow{\mathsf{tr}_i} \mathcal{V}^f(1^\lambda, 0^n, \rho, (b_j)_{j<i}) \\ \quad b_i := \mathcal{P}^f(1^\lambda, 0^n, \zeta, (a_j)_{j \leq i}) \\ \mathsf{d} \xleftarrow{\mathsf{tr}_\mathsf{k}} \mathcal{V}^f(1^\lambda, 0^n, \rho, (b_i)_{i \in [\mathsf{k}]}) \\ \mathsf{tr} := \mathsf{tr}_1 \parallel \cdots \parallel \mathsf{tr}_\mathsf{k} \end{array} \right] < p(n) \;\; .$$

We can show that

$$|Q^\star(f, w)| \geq t(n) - \frac{\mathsf{vq}(\lambda, n)}{p(n)} \;\; . \tag{12}$$

We define the completeness adversary $\mathcal{A}_{n,w}$ as follows:

$\mathcal{A}^f_{n,w}$: Output $(0^n, w)$.

We define $\mathsf{UF}_{n,w} := \{f \in \mathcal{O}_\lambda : f \in \mathsf{UF}_n \wedge (0^n, w) \in R_f\}$. We define $\mathsf{Flip}\,(f, w, i)$ as in the proof of Lemma 6.3.

Then, we can deduce from Eq. (12) and Definitions 7.2 and 7.3 that

$$\begin{aligned} &(1 - p(n)) \cdot \frac{t(n) - \frac{\mathsf{vq}(\lambda,n)}{p(n)}}{t(n)} \cdot \left(\left(1 - \frac{1}{2^{t(n)}}\right)^{2^{t(n)}-1} - \alpha(\lambda, n)\right) \\ &\leq \sum_{w \in \{0,1\}^{t(n)}} \Pr\left[\begin{array}{l} f \in \mathsf{UF}_{n,w} \\ \wedge\, \mathsf{d}' = 1 \end{array} \;\middle|\; \begin{array}{l} f \leftarrow \mathcal{O}_\lambda \\ (0^n, w) \leftarrow \mathcal{A}^f_{n,w} \\ i \leftarrow [t(n)] \\ f' := \mathsf{Flip}\,(f, w, i) \\ \rho \leftarrow \{0,1\}^{\mathsf{vr}} \\ \zeta \leftarrow \{0,1\}^{\mathsf{pr}} \\ \mathsf{d}' := \langle \mathcal{P}^f(1^\lambda, (0^n, w), \zeta), \mathcal{V}^{f'}(1^\lambda, 0^n, \rho) \rangle \end{array} \right] \end{aligned}$$

$$\leq \frac{\beta(\lambda, n, \mathsf{pq}, \mathsf{pt})}{t(n)} \ ,$$

a contradiction. $\square$

Acknowledgments

Ziyi Guan thanks Mika Göös and Dmitry Sokolov for pointing out relevant references. Ziyi Guan thanks Wenna Qin for helpful discussions at the early stage of this work. The authors thank Zijing Di, Giacomo Fenzi, Burcu Yıldız, and Zihan Yu for helpful comments on earlier drafts of this paper. Alessandro Chiesa and Ziyi Guan are partially supported by the Ethereum Foundation.

References

[AB18] Bariş Aydinlioğlu and Eric Bach. "Affine Relativization: Unifying the Algebrization and Relativization Barriers". In: *ACM Transactions on Computation Theory* 10.1 (2018).

[AIV92] Sanjeev Arora, Russell Impagliazzo, and Umesh Vazirani. *Relativizing versus nonrelativizing techniques: The role of local checkability.* Unpublished manuscript. 1992.

[AW09] Scott Aaronson and Avi Wigderson. "Algebrization: A New Barrier in Complexity Theory". In: *ACM Transactions on Computation Theory* 1.1 (2009), 2:1–2:54.

[BCCT13] 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. 2013, pp. 111–120.

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. "Interactive Oracle Proofs". In: *Proceedings of the 14th Theory of Cryptography Conference*. TCC '16-B. 2016, pp. 31–60.

[BCTV14] Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. "Scalable Zero Knowledge via Cycles of Elliptic Curves". In: *Proceedings of the 34th Annual International Cryptology Conference*. CRYPTO '14. Extended version at `http://eprint.iacr.org/2014/595`. 2014, pp. 276–294.

[BGS75] Theodore Baker, John Gill, and Robert Solovay. "Relativizations of the P =? NP Question". In: *SIAM Journal on Computing* 4.4 (1975), pp. 431–442.

[CCGOS23] 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 Theory and Application of Cryptographic Techniques*. EUROCRYPT '23. 2023, pp. 379–404.

[CCS22] Megan Chen, Alessandro Chiesa, and Nicholas Spooner. "On Succinct Non-interactive Arguments in Relativized Worlds". In: *Proceedings of the 41st Annual International Conference on Theory and Application of Cryptographic Techniques*. EUROCRYPT '22. 2022, pp. 336–366.

[CGSY23] Alessandro Chiesa, Ziyi Guan, Shahar Samocha, and Eylon Yogev. *Security Bounds for Proof-Carrying Data from Straightline Extractors*. Cryptology ePrint Archive, Paper 2023/1646. 2023.

[CL20] 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. 2020.

[COS20] Alessandro Chiesa, Dev Ojha, and Nicholas Spooner. "Fractal: Post-Quantum and Transparent Recursive Proofs from Holography". In: *Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT '20. 2020, pp. 769–793.

[CT10] 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*. ITCS '10. 2010, pp. 310–331.

[CY20] Alessandro Chiesa and Eylon Yogev. "Barriers for Succinct Arguments in the Random Oracle Model". In: *Proceedings of the 17th Theory of Cryptography Conference*. TCC '20. 2020, pp. 47–76.

[CY24] Alessandro Chiesa and Eylon Yogev. *Building Cryptographic Proofs from Hash Functions*. 2024. URL: `https://github.com/hash-based-snargs-book`.

[Cha+94] Richard Chang, Benny Chor, Oded Goldreich, Juris Hartmanis, Johan Håstad, Desh Ranjan, and Pankaj Rohatgi. "The random oracle hypothesis is false". In: *Journal of Computer and System Sciences* 49.1 (1994), pp. 24–39.

[Dek69] Michael Dekhtyar. "On the impossibility of eliminating exhaustive search in computing a function relative to its graph". In: *Doklady Akademii Nauk SSSR* 14 (1969), pp. 1146–1148.

[For94] Lance Fortnow. "The Role of Relativization in Complexity Theory". In: *Bulletin of the European Association for Theoretical Computer Science* 52 (1994), pp. 229–244.

[HN23] Mathias Hall-Andersen and Jesper Buus Nielsen. "On Valiant's Conjecture: Impossibility of Incrementally Verifiable Computation from Random Oracles". In: *Proceedings of the 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT '23. 2023.

[Hel86] Hans Heller. "On relativized exponential and probabilistic complexity classes". In: *Information and Control* 71.3 (1986), pp. 231–243.

[IKK09] Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. "An axiomatic approach to algebrization". In: *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*. STOC '09. 2009, pp. 695–704.

[KR09] Yael Tauman Kalai and Ran Raz. "Probabilistically Checkable Arguments". In: *Proceedings of the 29th Annual International Cryptology Conference*. CRYPTO '09. 2009, pp. 143–159.

[Lis86] Gerhard Lischke. "Oracle-Constructions to Prove All Possible Relationships Between Relativizations of P, NP, EL, NEL, EP and NEP." In: *Mathematical Logic Quarterly* 32.17-18 (1986), pp. 257–270.

[Mic00] Silvio Micali. "Computationally Sound Proofs". In: *SIAM Journal on Computing* 30.4 (2000). Preliminary version appeared in FOCS '94., pp. 1253–1298.

[Val08] Paul Valiant. "Incrementally Verifiable Computation or Proofs of Knowledge Imply Time/Space Efficiency". In: *Proceedings of the 5th Theory of Cryptography Conference*. TCC '08. 2008, pp. 1–18.