

Bounded variability of metric temporal logic

Carlo A. Furia¹  · Paola Spoletini²

Published online: 13 December 2016

© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract Deciding validity of Metric Temporal Logic (MTL) formulas is generally very complex and even undecidable over dense time domains; bounded variability is one of the several restrictions that have been proposed to bring decidability back. A temporal model has bounded variability if no more than v events occur over any time interval of length V , for constant parameters v and V . Previous work has shown that MTL validity over models with bounded variability is less complex—and often decidable—than MTL validity over unconstrained models. This paper studies the related problem of deciding whether an MTL formula has *intrinsic* bounded variability, that is whether it is satisfied *only* by models with bounded variability. The results of the paper are mainly negative: over dense time domains, the problem is mostly undecidable (even if with an undecidability degree that is typically lower than deciding validity); over discrete time domains, it is decidable with the same complexity as deciding validity. As a partial complement to these negative results, the paper also identifies MTL *fragments* where deciding bounded variability is simpler than validity, which may provide for a reduction in complexity in some practical cases.

Keywords Metric temporal logic · Bounded variability · Decidability and complexity

The first author's work was partially done at the Chair of Software Engineering of ETH Zurich, Switzerland. A preliminary version of this work appeared in the 21st International Symposium on Temporal Representation and Reasoning (TIME) in 2014 [23].

✉ Carlo A. Furia
furia@chalmers.se

Paola Spoletini
pspoleti@kennesaw.edu

¹ Department of Computer Science and Engineering, Chalmers University of Technology, Gothenburg, Sweden

² Department of Software Engineering and Game Development, Kennesaw State University, Kennesaw, GA, 30144, USA

Mathematics Subject Classification (2010) 03B70 · 03B44

1 Reaping the benefits of bounding variability

In mathematical logic, the fundamental decision problem is *validity*: establishing whether an arbitrary formula of the logic is satisfied by all possible models. Expressive logics whose validity problem is decidable, with tractable complexity, are powerful tools for the rigorous analysis of systems: verifying that system behavior satisfies a property amounts to checking the validity of the implication $S \Rightarrow P$, where S and P are formulas of the logic that formalize system and property.

For the analysis of real-time systems, whose behavior depends on the precise times in which events occur, *real-time logics* are the notation of choice. A popular real-time logic is the Metric Temporal Logic [3, 31], which is both expressive and intuitive. Unfortunately, validity is generally undecidable for MTL over dense time domains—precisely the most valuable semantics from the point of view of real-time modeling since a dense time domain can naturally accommodate truly asynchronous events over time. The only way out of this predicament is restricting the expressive power of MTL in exchange for a decidable validity problem.

One approach to restricting MTL expressiveness consists in introducing *semantic* constraints. We pursued this approach in previous work [18, 21, 22], where we showed that validity of MTL over dense time becomes decidable under the restriction of *bounded variability*. A nonnegative integer v and a time duration V identify all models with *variability bounded by v/V* : models where there are no more than v events (such as change of state) over any interval of length V . Models with bounded variability can still capture asynchronous behavior (events that are arbitrarily close in time) if the time domain remains dense. As we recall in Section 4, validity for MTL over dense time is fully decidable for models with variability bounded by v/V . Even over discrete time, where MTL is decidable without restrictions, deciding validity over models with bounded variability is often simpler in practice than the general case.

For an approach based on bounded variability to work satisfactorily in verification, it remains to be established whether models with bounded variability provide a sufficiently faithful formalization of system behavior. In some domains bounded variability naturally emerges as a physical property of the system being modeled. For example, the behavior of a digital event counter (such as the counter of ionization events in a Geiger-Müller tube) whose minimum time resolution is 0.01 seconds is captured without loss of precision whatsoever by models with variability bounded by 100 events over 1 second.

Guaranteeing that bounded variability is intrinsic may not be always feasible. A more analytical approach turns the issue of determining whether all models of an MTL formula have bounded variability into a decision problem distinct from validity. Given v and V , and an MTL formula ϕ , the *bounded variability problem* asks whether every model of ϕ has variability bounded by v/V . If the bounded variability problem is decidable, we can approach the study of validity as follows.

1. Build an MTL formula ϕ that formalizes the intended behavior under analysis.
2. Check whether ϕ has intrinsic bounded variability—that is, whether every model of ϕ has variability bounded by v/V .

3. If the check in (2) is successful, analyze the validity of ϕ using the decidable algorithms for models with bounded variability. This is simpler than deciding validity for general models, and it does not miss any possible behavior.
4. If the check in (2) is unsuccessful, then be aware that the bounded variability constraint misses some possible behaviors, and hence a different, possibly more powerful, approach to analyzing ϕ is needed.

For this approach to work, the bounded variability problem has to be decidable and has to have a lower complexity than the validity problem for MTL under the same interpretation.

The main contribution of this paper is an analysis of the decidability and complexity of the bounded variability problem for MTL under various interpretations, including both discrete and dense time domains. The analysis leverages a connection between the bounded variability problem and various decision problems involving overflows of counters in a variant of Minsky’s counter machines [38]. The results, summarized in Table 1, are mainly negative: the bounded variability problem is generally undecidable over dense time, whereas it is decidable over discrete time but as complex as validity. These results denote major hurdles to pursuing the idea of finding formulas with bounded variability in order to use simpler algorithms for validity: the complexity of solving the bounded variability problem (step 2 above) dominates and nullifies the benefits of using simplified algorithms.

As a partial converse of these negative results, Section 7 identifies two MTL *fragments*, one over discrete time and one over dense time, for which the above approach is feasible. The bounded variability problem is fully decidable for formulas in such “bounded friendly” fragments, which may provide enough expressiveness to formalize system behavior in interesting special cases.

Outline Section 1.1 summarizes related work in real-time formalisms, focusing on the restrictions—syntactic and semantic—that have been proposed to obtain a decidable validity problem. Section 2 formally introduces fundamental notions used throughout the paper, including those of timed word and bounded variability. Section 3 presents MTL in detail: its syntax (Section 3.1), its semantics under different interpretations (Section 3.2), and the decidability and complexity of validity under each interpretation (Section 3.3). Section 4 summarizes and combines our previous work on the decidability and complexity of MTL validity over models with bounded variability—for discrete as well as dense time domain. Section 5 recalls the definition of nondeterministic n -counter machines, introduces three decision problems involving counter overflows (Section 5.1) and studies their complexity,

Table 1 Overview of the main results of this paper; all complexity bounds in the table are tight, that is the problems are *complete* for the given complexity classes

| TIME DOMAIN | VALIDITY PROBLEM | | BOUNDED VARIABILITY PROBLEM | |
|--------------------|---------------------|-------------------|-----------------------------|-------------------|
| | <i>decidability</i> | <i>complexity</i> | <i>decidability</i> | <i>complexity</i> |
| discrete | decidable | EXPSPACE | decidable | EXPSPACE |
| dense (continuous) | undecidable | Σ_1^1 | undecidable | coRE |

The bounded variability problem for MTL is undecidable over dense time (with continuous semantics), but its complexity (**coRE**-complete) is lower than the complexity of the validity problem under the same semantics (Σ_1^1 -complete); the bounded variability problem for MTL is decidable over discrete time, and its complexity (**EXPSPACE**-complete) is the same as the complexity of the validity problem under the same semantics. See Tables 2 and 4 for details

and illustrates the connection between MTL and such counter machines (Section 5.2). Section 6 presents the main results of the paper. After formally defining two variants of the bounded variability problem (Section 6.1), it proves the complexity of the two problems for various MTL semantics over both dense (Section 6.2) and discrete (Section 6.3) time. Section 7 provides two examples of MTL fragments, again over both discrete (Section 7.1) and dense (Section 7.2) time, for which the bounded variability problem is tractable—fragments that we dub “bounded friendly”. Section 8 concludes with a brief overview of the results and a summary of open problems that belong to future work.

1.1 Related work

The amount of work on temporal logic and its decidability problems is staggering, and stretching back over four decades [15, 17]. This section focuses on the research that is most directly related to the paper’s techniques and results, and in particular on the logic MTL and its variants.

1.1.1 Linear temporal logic

Linear-temporal logic (LTL)—introduced in computer science by Pnueli [50]—is arguably the most widely known kind of temporal logic. The expressiveness, decidability, and complexity properties of LTL are remarkably robust with respect to variants in the underlying semantics—in particular whether time is a discrete or dense set. The bulk of the research has focused on the infinite discrete-time semantics of LTL, where the expressiveness [25] and **PSPACE**-completeness [59] of LTL are well known, and where the correspondence with Büchi automata [60, 61] supported numerous theoretical and practical developments. However, most of the results over discrete time generalize to other semantics; for example, Kamp’s seminal work on the expressive completeness of LTL [30] applies to all Dedekind-complete sets (including the real numbers) as time domain, and Reynolds extended the **PSPACE**-completeness result to the reals [54]. The complexity of LTL fragments and variants has also been extensively surveyed [5, 12, 33]. Some of these latter results are recalled and used in Section 7.

1.1.2 Metric temporal logic

The picture is considerably more fragmented if we consider temporal logics for real time, that is equipped with a metric on time. Syntactically, the most natural extension of LTL to real time is the Metric Temporal Logic (MTL), which is the focus on the present paper. MTL was introduced by Koymans [31] as a metric first-order temporal logic supporting unrestricted quantification over time variables. It was Alur and Henzinger’s subsequent work [3], however, that popularized the propositional subset of Koyman’s MTL; since their seminal work, “MTL” refers to this propositional subset, where the temporal operators of LTL are decorated with quantitative bounds on time in the form of intervals. As we recall in Section 3.3, Alur and Henzinger showed that MTL is fully decidable over discrete time domains, where it boils down to an exponentially succinct version of LTL. They also showed that MTL becomes highly undecidable over dense time domains—in particular, the nonnegative reals.

Decidable MTL fragments The undecidability proof of MTL over dense time opened a new line of research consisting in devising *restrictions* that get back decidability while

preserving as much expressive power as possible. A number of works targeted *syntactic* restrictions, that is they considered fragments of all MTL formulas. A notable result in this line of work is the logic MITL [2], which prohibits *punctual* (or *singular*) intervals in temporal operators. With this restriction, MITL is fully decidable and **EXSPACE**-complete, matching the complexity of MTL over discrete time; the original automata-based decision procedure [2] was simplified in a later series of works [34–36, 42]. Bouyer et al. showed different syntactic restrictions to MTL that achieve decidability (typically with **EXSPACE** complexity) while allowing punctual intervals [6]; for example, the MTL fragment where all time intervals are bounded, that is have integer endpoints, is fully decidable.

Hirshfeld and Rabinovich approached the problem of devising decidable dense-time temporal logics from a different angle. Instead of extending the automata-theoretic methods that are mostly used over discrete time, they revisited the connection between temporal logic and monadic logic [24] and came up with a fully-decidable dense-time logic [26] which is as expressive as MITL. Their approach also paved the way for pushing the expressiveness of the logic while preserving decidability [27, 51].

Ouaknine and Worrell also pushed the limits of MTL decidability over dense time, by proving that full MTL is decidable under a specific dense-time semantics known as the finite-word pointwise semantics [45] (see Section 3.2 for a definition). Their result has little practical impact, since MTL is just “barely” decidable over this semantics (that is, it is decidable but has daunting non-primitive recursive complexity), but was quite surprising and convincingly showed that the properties of MTL over dense time are considerably more complicated than over discrete time [29].

Decidable semantic restrictions to MTL Ouaknine and Worrell also considered *semantic* restrictions to MTL over dense time, most notably the restriction to *bounded time*, where there is a finite known upper bound T on the maximum time of occurrence of events in any behavior. Time-bounded MTL validity is decidable [43, 47], with the—usual, for dense-time verification—complexity of **EXSPACE**.

This paper considers a different kind of semantic restriction, called *bounded variability*, which limits the maximum number of events that can occur within a time unit. Wilke first studied bounded variability for timed automata and monadic logic [62]; his results imply the decidability of MTL over dense time with bounded variability. Previous work of ours studied the complexity of MTL with bounded variability [18, 20], and extended some of the techniques to the case of discrete time [21, 22], where we showed how the LTL validity problem can be simplified under the assumption that only $v < V$ change events happen every V discrete time steps. Section 4 recalls in detail these works of ours, and connects them to the present paper’s results.

1.1.3 Other real-time logics

Bounded variability for other notations Bounded variability can be considered a natural semantic restriction over dense time, in that it accommodates asynchrony (events can occur arbitrarily close in time) while constraining the maximum “speed” or “density” of state-change events. In fact, it has been considered for formalisms other than MTL, including timed automata [62], duration calculus [16], and different kinds of real-time temporal logics [55, 56]

Interval temporal logic The study of real-time is somewhat less developed in temporal logics not based on LTL, and especially in temporal logics based on intervals rather

than points. In fact, interval temporal logics are highly expressive, often undecidable formalisms whose detailed classification has occupied researchers up until recent years [7]. Only recently have real-time interval temporal logics been studied in some detail [9, 39]. Some interval logics, such as propositional neighborhood logic, remain decidable over the reals [41]. Other variants, such as a fragment of Halpern-Shoham logic that includes the two Allen’s relations “meets” and “begins” is decidable over the rationals but undecidable over the reals [40]. The classic duration calculus includes fragments that are decidable over dense time [10, 11, 16, 53].

1.1.4 Background

The undecidability results of Section 6.2 use reductions from undecidable problems of non-deterministic n -counter machines, which we introduce in Section 5. These are a kind of Minsky’s counter machines [38]; their connection with MTL was first exploited by Alur and Henzinger [3].

We assume readers are familiar with fundamental notions of computability and complexity theory. In particular, the arithmetical hierarchy [57], the complexity classes **NP**, **EXP**, **PSPACE**, and **EXSPACE**, and the notion of (polynomial-time) reduction between decision problems [48].

2 Timed words and bounded variability

Time domains: discrete and dense Complexity and other properties of temporal logics change significantly according to whether the time domain is a discrete or a dense set. The results of this paper will target both kinds of time domain: the *discrete* set of the nonnegative integers \mathbb{N} and the *dense* (and continuous) set of the nonnegative reals $\mathbb{R}_{\geq 0}$. Definitions applicable to both domains will use the symbol \mathbb{T} to denote an unspecified time domain that stands for either \mathbb{N} or $\mathbb{R}_{\geq 0}$. Generalizing the results to so-called bi-infinite [49, Chap. 9] time domains (such as \mathbb{Z} and \mathbb{R}) is possible in some cases but is outside the scope of the present paper.

Timed words Timed words model real-time behavior as sequences of instantaneous events that occur over time at precise instants. Since they are natural generalizations of (untimed) words—the canonical models of behavior in qualitative temporal logic [17, 60]—timed words are widely-used models that account for a variety of real-time formalisms. In particular, they are standard for MTL since Alur and Henzinger’s work [2–4]; for this reason, they are used throughout the present paper.

A *timed word* over propositions \mathcal{P} and time domain \mathbb{T} is a sequence

$$w = (P_0, t_0) (P_1, t_1) \cdots \tag{1}$$

of pairs $(P_k, t_k) \in (2^{\mathcal{P}} \times \mathbb{T})$ such that:

1. Nonnegative integers $k \geq 0$ denote *position* in the timed word w .
 - (a) *Finite word*: If $w \in (2^{\mathcal{P}} \times \mathbb{T})^*$, then there exists a nonnegative integer, denoted by $|w|$ and called the *length* of w , such that only positions $0 \leq k < |w|$ are *valid*.
 - (b) *Infinite word*: If $w \in (2^{\mathcal{P}} \times \mathbb{T})^\omega$, then every position $0 \leq k$ is valid, and we let $|w| = \infty$ and $t_{|w|-1} = \infty$.

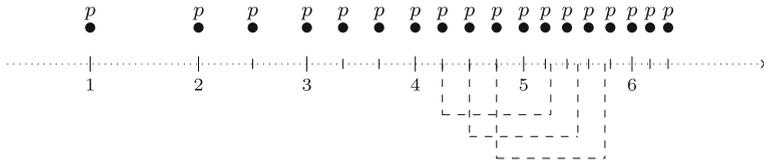


Fig. 1 The timed word u , introduced in Example 1, has finite but unbounded variability

2. For each valid position k : P_k is a nonempty¹ subset of \mathcal{P} denoting the propositions holding (or events occurring) at position k ; and $t_k \in \mathbb{T}$ is a timestamp denoting the absolute time of occurrence of the events at position k .
3. *Monotonicity*: the timestamp sequence is strictly monotonic, that is, if h and k are two valid positions in w such that $h < k$, then $t_h < t_k$.
4. *Finite variability*: the timestamp sequence is finite or divergent, that is, either w is finite or, for all $t \in \mathbb{T}$, there exists a valid position k such that $t_k > t$.

Finite variability (also called non-Zenoness [1, 26]) is a customary requirement for infinite words over dense time domains, where it rules out pathological behaviors with infinitely many events occurring over a finite amount of time. It holds trivially over finite words, and it follows from monotonicity over discrete-time infinite words.

5. *Base time*: the first timestamp t_0 , if it exists, is 0. This is merely a technical convenience to more reconcile the two MTL semantics (pointwise and continuous) introduced in Section 3.2.

Bounded variability of timed words Finite variability prevents timestamps in a timed word from accumulating at a finite time, but it does not introduce a word-wide maximum density of timestamps.

Example 1 Consider a timed word $u \in (\{p\} \times \mathbb{R}_{\geq 0})^\omega$ such that, for all integers $y > 0$ and $0 \leq x < y$, $y + x/y$ is a timestamp t_k in u for some $k > 0$; that is, u 's timestamp sequence is the one shown in Fig. 1:

$$0, 1, 2, 2 + \frac{1}{2}, 3, 3 + \frac{1}{3}, 3 + \frac{2}{3}, 4, 4 + \frac{1}{4}, 4 + \frac{2}{4}, 4 + \frac{3}{4}, 5, \dots$$

Note that u has $n + 1$ occurrences of p between time n and time $n + 1$ (included). In fact, Lemma 2 shows that *any* interval between time t and time $t + 1$ in u contains up to $\lfloor t \rfloor + 1$ valid positions (or timestamps), each with an occurrence of p . Thus, even if u has finite variability (the timestamps grow indefinitely large), there is no bound on the number of events that occur over a unit time interval.

Lemma 2 For all $0 < \tilde{t} \in \mathbb{R}_{\geq 0}$, timed word u —defined in Example 1—includes up to $\lfloor \tilde{t} \rfloor + 1$ valid positions with timestamps between \tilde{t} and $\tilde{t} + 1$ (included).

Proof Without loss of generality, let \tilde{t} be a timestamp in u greater than 1. Such \tilde{t} can be written as $y + x/y$ for integers $0 \leq x < y$ and $y = \lfloor \tilde{t} \rfloor$; the interval from \tilde{t} to $\tilde{t} + 1$ includes all instants up to some $(y + 1) + x'/(y + 1)$, for integers $0 \leq x' \leq y$. There are exactly

¹Without loss of generality, since one can always introduce a special letter ϵ denoting absence of events.

$(y + 1 - x) + x'$ instants over such interval: $(y + 1) - x$ instants between \tilde{t} (included) and $y + 1$ (included), and x' instants between $y + 1$ (excluded) and $(y + 1) + x'/(y + 1)$ (included). The constraint that $\tilde{t} + 1 \geq (y + 1) + x'/(y + 1)$ —so that the unit interval is indeed the one including $(y + 1) + x'/(y + 1)$ —is equivalent to $0 \leq x' \leq x < y$. Thus, picking $x = x'$ maximizes the number $(y + 1 - x) + x'$ of instants in the interval, which becomes just $y + 1$. □

A timed word w has *variability bounded by v/V* —read “ v over V ”—for $0 < V \in \mathbb{T}$ and $v \in \mathbb{N}$, iff it has at most v valid positions within any time interval of length V . That is, for all positions k such that $k + v < |w|$,

$$t_{k+v} - t_k > V.$$

The symbols $\mathcal{B}_{\mathbb{T}}^*[v/V]$ and $\mathcal{B}_{\mathbb{T}}^\omega[v/V]$ respectively denote the sets of all finite and of all infinite timed words over \mathbb{T} with variability bounded by v/V . In contrast, the symbols $\mathcal{B}_{\mathbb{T}}^*[\infty]$ and $\mathcal{B}_{\mathbb{T}}^\omega[\infty]$ respectively denote the sets of all finite and of all infinite timed words over \mathbb{T} without restriction of bounded variability—as emphasized by the ∞ symbol.

Remark 3 Any given *finite* timed word $w \in \mathcal{B}_{\mathbb{T}}^*[\infty]$ has variability bounded by *some* v/V ; precisely by $|w| / \max(1, (t_{|w|-1} - t_0))$. However, the notion of bounded variability can still be nontrivial when applied to infinite sets of finite timed words.

For example, consider the denumerable set $U = \{u_n \mid 1 < n \in \mathbb{N}\}$, where each u_n is obtained by truncating u in Example 1 up to and including the integer timestamp with value n . It follows from Example 2 that each $u_n \in U$ is bounded by $n/1$. However, there is no finite \bar{n} such that all words in U are bounded by \bar{n} , because there is always a longer word in U with more (albeit finitely many) than \bar{n} valid positions over a time interval of unit length.

3 Metric temporal logic

Metric Temporal Logic (MTL) naturally extends classic Linear Temporal Logic (LTL) by providing means to express quantitative constraints on the time of occurrence of events. This section presents the syntax and semantics of propositional MTL.

Time intervals MTL decorates LTL temporal operators with intervals of the time domain \mathbb{T} . An *interval* is a convex subset of \mathbb{T} written as $I = \langle a, b \rangle$, where a and b are elements of $\mathbb{T} \cup \{\infty\}$, \langle is one of $($ and $[$, and \rangle is one of $)$ and $]$. Such an interval I consists of all elements of \mathbb{T} between a and b , where a is included iff \langle is $[$, and b is included iff \rangle is $]$. We normally consider intervals with integer endpoints a and b even if the time domain \mathbb{T} is $\mathbb{R}_{\geq 0}$ (but the interval itself includes all real numbers between a and b if and only if $\mathbb{T} = \mathbb{R}_{\geq 0}$). For $s \in \mathbb{T}$, the pseudo-arithmetic expressions $> s, \geq s, < s, \leq s$, and $= s$ are abbreviations for the intervals $(s, \infty), [s, \infty), [0, s), [0, s]$ and $[s, s]$.

3.1 Syntax of MTL

MTL formulas are built out of one binary temporal operator, the *until* \mathbf{U}_J , and the usual propositional logic connectives, according to the recursive definition:

$$\phi ::= \top \mid p \mid \neg\phi_1 \mid \phi_1 \wedge \phi_2 \mid \mathbf{U}_J(\phi_1, \phi_2),$$

where p ranges over the alphabet of propositions \mathcal{P} , J is an interval of the time domain \mathbb{T} with integer endpoints, and \top represents the Boolean constant for “true”.

Derived operators The definitions of \perp (the Boolean constant for “false”), \vee , \Rightarrow , and \Leftrightarrow are as in standard propositional logic. The symbol α abbreviates the formula $\bigvee_{p \in \mathcal{P}} p$, which holds iff at least one proposition in \mathcal{P} holds. The derived temporal operators we use in this paper and their definitions are:

| SYMBOL | DEFINITION | NAME |
|---------------------------------|--|--------------------|
| $\diamond_J(\phi)$ | $\bigcup_J(\top, \phi)$ | eventually |
| $\square_J(\phi)$ | $\neg \diamond_J(\neg \phi)$ | globally or always |
| $\widehat{U}_J(\phi_1, \phi_2)$ | $\bigcup_J(\alpha \Rightarrow \phi_1, \phi_2)$ | action until |
| $\bigcirc_J(\phi)$ | $\widehat{U}_J(\perp, \phi)$ | next |

All operators are standard except the *action until*, which is a variant of the usual until operator where the first argument has to hold only if α hold—that is only if any proposition holds. The action until is useful to accommodate timed word semantics, where no proposition holds between consecutive elements in any word.

Operator precedence is as follows: \neg has the highest precedence, then \wedge , then \vee , then \Rightarrow , then all temporal operators, and finally \Leftrightarrow . We may omit parentheses around arguments of temporal operators when unambiguous, and drop intervals of the form $[0, \infty)$ to stress the correspondence with LTL (where all temporal operators have an implicit interval $[0, \infty)$).

Size of formulas The size $|\phi|$ of an MTL formula ϕ is the size of its encoding as a string, using a succinct (say, binary) encoding of the constants used as endpoints of time interval. Since we are going to be concerned only with asymptotic complexity, the details of the encoding are irrelevant.

3.2 Semantics of MTL

Pointwise vs. continuous semantics There are two main semantics of MTL [6, 13]: the pointwise semantics and the continuous semantics. If the time domain is the discrete set \mathbb{N} , the differences between the two semantics are practically inconsequential. In contrast, whether a pointwise or a continuous semantics is chosen affects fundamental properties of decidability of MTL over dense time $\mathbb{R}_{\geq 0}$, and correspondingly some of the results of the present paper.

Pointwise semantics Given an MTL formula ϕ , a timed word w as in (1), and a position $k \in \mathbb{N}$, we write $w, k \models_p \phi$ to denote that w is a model of ϕ at position k under the pointwise semantics. In this context, k is sometimes referred to as the “current” position or step. The *pointwise* satisfaction relation \models_p is inductively defined as follows:

| | | |
|--|-----|---|
| $w, k \models_p \top$ | iff | $k < w $; |
| $w, k \models_p p$ | iff | $p \in P_k$; |
| $w, k \models_p \neg \phi_1$ | iff | $w, k \not\models_p \phi_1$; |
| $w, k \models_p \phi_1 \wedge \phi_2$ | iff | $w, k \models_p \phi_1$ and $w, k \models_p \phi_2$; |
| $w, k \models_p \bigcup_J(\phi_1, \phi_2)$ | iff | there exists a position $k < h < w $ such that: $t_h - t_k \in J$, $w, h \models_p \phi_2$, and, for all positions $k < x < h$, $w, x \models_p \phi_1$; |
| $w \models_p \phi$ | iff | $w, 0 \models_p \phi$. |

It is useful to spell out the semantics of the derived *next* operator:

$$w, k \models_p \bigcirc_J(\phi_1) \text{ iff } k + 1 < |w|, t_{k+1} - t_k \in J, \text{ and } w, k + 1 \models_p \phi_1.$$

That is, $w, k \models_p \bigcirc_J(\phi_1)$ holds iff ϕ_1 holds at the next position in w at a time whose difference with the current evaluation time t_k is in J .

Remark 4 The semantics of \mathbf{U} and $\widehat{\mathbf{U}}$ coincide under the pointwise semantics, since formulas are only evaluated at valid positions, where α invariably holds (as $P_k \neq \emptyset$ at all valid positions k).

Continuous semantics Given an MTL formula ϕ , a timed word w as in (1), and a time instant $t \in \mathbb{T}$, we write $w, t \models_c \phi$ to denote that w is a model of ϕ at time t under the continuous semantics. In this context, t is sometimes referred to as the “current” time. The *continuous* satisfaction relation \models_c is inductively defined as follows:

| | | |
|---|-----|---|
| $w, t \models_c \top$ | iff | $t \leq t_{ w -1}$; |
| $w, t \models_c p$ | iff | there exists a position $k \in \mathbb{N}$ such that: $k < w , t_k = t$, and $p \in P_k$; |
| $w, t \models_c \neg\phi_1$ | iff | $w, t \not\models_c \phi_1$; |
| $w, t \models_c \phi_1 \wedge \phi_2$ | iff | $w, t \models_c \phi_1$ and $w, t \models_c \phi_2$; |
| $w, t \models_c \mathbf{U}_J(\phi_1, \phi_2)$ | iff | there exists $u > t$ such that: $u - t \in J$, $w, u \models_c \phi_2$, and, for all $t < v < u$, $w, v \models_c \phi_1$; |
| $w \models_c \phi$ | iff | $w, 0 \models_c \phi$. |

Remark 5 The continuous semantics of the derived *next* operator $w, t \models_c \bigcirc_J(\pi)$, when π is a purely propositional formula, requires that there exists a valid position $h \in \mathbb{N}$ such that $t < t_h$, there are no valid positions with timestamp between t and t_h , $t_h - t \in J$, and $w, t_h \models_c \pi$. This corresponds closely to the semantics of *next* under the pointwise semantics, since the semantics of the *action until* (on which *next*’s definition relies) forces the evaluation of π in the *next* valid position from the current t . We will make use of this correspondence to match results in the pointwise and in the continuous semantics.

Example 6 The MTL formulas $\phi_{1+1} = \diamond_{=1}\diamond_{=1} p$ and $\phi_2 = \diamond_{=2} p$ are equivalent under the continuous semantics but not under the pointwise semantics. To illustrate the difference, consider the finite word $w_2 = (\{p\}, 0) (\{p\}, 2)$. Clearly, $w_2 \models_c \phi_2$ and $w_2 \models_c \phi_{1+1}$ — w_2 is a model of both ϕ_2 and ϕ_{1+1} under the continuous semantics, where both formulas only require p to occur at time 2. In contrast, $w_2 \models_p \phi_2$ but $w_2 \not\models_p \phi_{1+1}$ — w_2 is a model of ϕ_2 but not of ϕ_{1+1} under the pointwise semantics, where ϕ_{1+1} requires an intermediate position with timestamp 1 to satisfy its nested eventually operators.

3.3 Decidability and complexity of MTL

Interpretations An *interpretation* I is defined by a quadruple

$$I = \langle \mathbb{T}, \circ, \mathbb{V}, \mathbf{s} \rangle, \tag{2}$$

consisting of a time domain \mathbb{T} , a choice \circ of finite ($*$) or infinite (ω) words, a choice \mathbb{V} of bounded (v/V) or finite but possibly unbounded (∞) variability, and a choice \mathbf{s} of pointwise (p) or continuous (c) semantics. We use symbols of the form $\mathcal{B}_{\mathbb{T}}^{\circ, \mathbb{V}}[\mathbb{V}](\phi)_{\mathbf{s}}$ to denote sets of timed words that satisfy ϕ under a certain interpretation. For example, $\mathcal{B}_{\mathbb{T}}^*[\infty](\phi)_p$ is

the set of all finite timed words over \mathbb{T} that satisfy ϕ under the pointwise semantics; and $\mathcal{B}_{\mathbb{T}}^*[v/V](\phi)_c$ is the set of all timed words over \mathbb{T} with variability bounded by v/V that satisfy ϕ under the continuous semantics.

Validity and satisfiability An MTL formula ϕ is *satisfiable* under an interpretation $I = \langle \mathbb{T}, \circ, \mathbb{V}, \mathfrak{s} \rangle$ if $\mathcal{B}_{\mathbb{T}}^\circ[\mathbb{V}](\phi)_s \neq \emptyset$; otherwise ϕ is *unsatisfiable*. It is *valid* if $\mathcal{B}_{\mathbb{T}}^\circ[\mathbb{V}](\phi)_s = \mathcal{B}_{\mathbb{T}}^\circ[\mathbb{V}]$; otherwise ϕ is *invalid*. Since MTL formulas are closed under complement, satisfiability and validity are dual under complement: ϕ is valid if and only if $\neg\phi$ is unsatisfiable.

MTL decidability MTL is *decidable* under an interpretation I if deciding whether an arbitrary formula ϕ is satisfiable under I is a decidable problem. The “complexity of MTL under interpretation I ” refers to the complexity of satisfiability or validity under I .² The central columns of Table 2 recall the known complexity results for MTL validity under different interpretations, without the assumption of bounded variability.

MTL is decidable over *discrete time* over both finite and infinite words [3], and under both the pointwise and the continuous semantics (even if most authors use infinite words and pointwise semantics [4], extending their results to the other interpretations is straightforward). In all cases, satisfiability is complete for **EXPSpace**, which in practice determines algorithms that run in doubly-exponential time in the worst case. This is an exponential more than LTL, which is **PSpace**-complete; the succinct binary encoding of interval endpoints in MTL entirely accounts for this exponential blow-up—a manifestation of the “succinctness phenomenon” [48, Chap. 20].

MTL is invariably more complex over *dense time*, but the exact complexity depends on the specific interpretation chosen. Under the continuous semantics, MTL is highly undecidable—precisely, validity is Σ_1^1 -complete [3]. Under the pointwise semantics, it is still undecidable over infinite words [44]—albeit with a lower undecidability degree than over infinite words—whereas it becomes decidable over finite words [45]. However, its astronomical complexity (non-primitive recursive) makes this decidability result of little practical significance. A related result is that, over dense time, the continuous semantics is strictly more expressive than the pointwise semantics [14].

4 Metric temporal logic with bounded variability

The decidability and complexity of MTL under different interpretations change significantly if one considers models with bounded variability, as can be seen by comparing the central to the rightmost columns in Table 2.

4.1 Expressing bounded variability in temporal logic

Some of the constructions in this paper—and, in particular, the proofs of the upper bounds of deciding MTL over models with bounded variability—rely on being able to express that a model has bounded variability syntactically by means of MTL formulas. Precisely,

²If satisfiability (resp. validity) belongs to a complexity class that is closed under complement, validity (resp. satisfiability) belongs to the same complexity class, and hence we can speak generically of “complexity of MTL” without specifying whether we refer to satisfiability or validity. In the general case, satisfiability is in some complexity class **C** iff validity is in the complement class $\overline{\mathbf{C}}$.

Table 2 Known complexity results for MTL validity under different interpretations

| \mathbb{T} | \circ | s | W/ FINITE VARIABILITY | | | W/ v/V BOUNDED VARIABILITY | | |
|-----------------------|----------|--------|-----------------------|---------------------------|-----------------|------------------------------|---------------------------|-------------|
| | | | ω | ρ/c | DEC? COMPLEXITY | DEC? | COMPLEXITY | |
| \mathbb{N} | | | Y | EXPSpace -complete | [3] | Y | EXPSpace -complete | Section 4.2 |
| $\mathbb{R}_{\geq 0}$ | * | ρ | Y | non-primitive recursive | [45] | Y | EXPSpace -complete | Section 4.3 |
| $\mathbb{R}_{\geq 0}$ | ω | ρ | N | coRE -complete | [44] | Y | EXPSpace -complete | Section 4.3 |
| $\mathbb{R}_{\geq 0}$ | | c | N | Σ_1^1 -complete | [3] | Y | EXPSpace -complete | Section 4.3 |

For each choice of time domain \mathbb{T} , finite (*) or infinite (ω) timed words, and pointwise or continuous semantics, the table shows whether MTL is DECIDABLE (yes or no) under that interpretation and the complexity of deciding validity, both for models with FINITE variability and for models with variability BOUNDED by v/V . Blank cells correspond to parameters whose choice does not affect the result of the row. Next to each complexity result there is a reference—to another paper or to a section of the present paper—where it is proved

expressing bounded variability *succinctly* requires two variants of MTL that we call MTL^+ and MITL^+ .

MTL^+ augments MTL with the temporal operators B_T^n , for $n \in \mathbb{N}$ and a positive $V \in \mathbb{T}$. B_T^n holds iff there are at most n valid positions within the next T time units; that is:³

$$\begin{aligned}
 w, k \models_p B_T^n & \quad \text{iff} \quad k + n \geq |w| & \quad \text{or} & \quad t_{k+n} - t_k > T; \\
 w, t \models_c B_T^n & \quad \text{iff} \quad k + n \geq |w| & \quad \text{or} & \quad t_{k+n} - t_k > T,
 \end{aligned}$$

where k is the smallest position such that $t \leq t_k$.

MITL^+ is MTL^+ where all intervals appearing in temporal operators are non-singular (that is, include more than one element); equivalently, MITL^+ is MITL [2] augmented with the temporal operators B_T^n .

Based on previous results, it is possible to establish tight bounds on the complexity of MTL^+ and MITL^+ .

Proposition 7 *Satisfiability of MTL^+ over discrete-time models is **EXPSpace**-complete.*

Proof (idea) For the upper bound, it is possible to minimally modify the decision procedure for TPTL [4] so that operators with the semantics of B_T^n are handled natively without complexity penalty. This is enough since TPTL is as complex, and as expressive, as MTL over discrete time [3].

For the lower bound, note that every MTL formula is also an MTL^+ formula, and MTL satisfiability is **EXPSpace**-hard over discrete time [3]. □

Proposition 8 *Satisfiability of MITL^+ over dense-time models is **EXPSpace**-complete.*

Proof (idea) For the upper bound, it is possible to encode B_T^n in Hirshfeld and Rabinovich’s TLCI [27, 51, 52] as $\Box(\neg C_{n+1}^{[0, T]}\top)$, where $C_k^{[a, b]}\phi$ is TLCI’s counting operator—which expresses that ϕ holds at least k times over interval $[a, b]$. (While Hirshfeld and

³Note that “the smallest position k such that $t \leq t_k$ ” is well defined over words with *finite* variability [17, Sec. 9.4.1].

Rabinovich [27, 51, 52] define the counting operators only for open intervals, closed intervals can be indirectly encoded along the lines of our previous work [18]). Satisfiability for TICI is **PSPACE**-complete under unary encoding [52]; hence MITL⁺ is in **EXPSpace** assuming a succinct encoding of constants.

For the lower bound, note that every MITL formula is also an MITL⁺ formula, and MITL satisfiability is **EXPSpace**-hard over dense time [2]. □

Expressing bounded variability syntactically is straightforward using MITL⁺.

Lemma 9 *Given an interpretation $\langle \mathbb{T}, \circ, v/V, \mathbf{s} \rangle$, one can build, in polynomial time, an MITL⁺ formula $\mathcal{V}_{v,V}$ such that: (1) $\mathcal{B}_{\mathbb{T}}^{\circ}[v/V](\mathcal{V}_{v,V})_{\mathbf{s}} = \mathcal{B}_{\mathbb{T}}^{\circ}[\infty](\mathcal{V}_{v,V})_{\mathbf{s}}$ (that is, $\mathcal{V}_{v,V}$ is satisfied precisely by all words with variability bounded by v/V under the given interpretation); and (2) $\mathcal{V}_{v,V}$ has size polynomial in $\log v$ and $\log V$.*

Proof It is clear that MITL⁺ formula $\mathcal{V}_{v,V} = \Box(\mathbf{B}_v^v)$ satisfies the lemma’s statement. □

Remark 10 Adding the operators \mathbf{B}_T^a does not affect expressiveness or complexity, but provides additional succinctness. In fact, it is possible to build an MTL formula whose models are the same as the models of $\mathcal{V}_{v,V}$ (defined in Section 9) modulo projection of extra propositions; however, since MTL does not have compact operators such as \mathbf{B}_V^v , this would introduce an exponential blow-up in $\log v$. To demonstrate, assume a discrete time model, and consider $v > 0$ propositions p_i , for $i = 1, \dots, v$. Proposition p_1 has to hold initially, followed by p_2, \dots, p_v in sequence; the sequence repeats indefinitely:

$$B_v = p_1 \wedge \bigwedge_{1 \leq k \leq v} \left(\Box(p_1 \Leftrightarrow \bigcirc p_{k \oplus 1}) \wedge \Box \left(p_k \Rightarrow \bigwedge_{1 \leq h \neq k \leq v} \neg p_h \right) \right),$$

where $a \oplus b$ is a shorthand for $1 + ((a + b) \bmod v)$. Since every p_k holds in a different position, we can express bounded variability by requiring that the timestamp of the next $(v + 1)$ -th position in the future be greater than V with respect to the current position’s (and note that $k \oplus v = k$):

$$B_{v,V} = \bigwedge_{1 \leq k \leq v} \Box(p_k \Rightarrow U_{>V}(\neg p_k, p_k)).$$

Thus, $B_v \wedge B_{v,V}$ is satisfied only by models with variability bounded by v/V . However, $B_{v,V}$ has size exponential in the size of the instance of $\mathbf{BV}_{v/V}(\mathbb{N}, \omega, \mathbf{p})(\phi)$, precisely because $\mathbf{BV}_{v/V}(\mathbb{N}, \omega, \mathbf{p})(\phi)$ encodes v succinctly whereas $B_{v,V}$ enumerates v propositions. Encoding the modulo- v counter in binary (using $n = \lfloor \log_2 v \rfloor + 1$ propositions) does not seem to help: while updates to the counter itself can be done with formulas of size polynomial in n , expressing in MTL the fact that the timestamp of the “next” occurrence is greater than V (with respect to the current position’s) seems to require enumerating all $2^n = v$ values for the counter. Similar considerations would demonstrate that MITL⁺ is more succinct than MITL.

4.2 Bounded variability over discrete time

The asymptotic complexity of deciding MTL is the same over models with bounded variability as it is over finitely variable ones, that is satisfiability is **EXPSpace**-complete.

The lower bound is easy to prove because every word over discrete time has variability trivially bounded by $2/1$ —which is tantamount to monotonicity of timestamps. Therefore, satisfiability over discrete-time models trivially reduces to satisfiability over discrete-time models with bounded variability, which establishes that the latter problem is **EXPSPACE**-hard.

The upper bound follows from Lemma 9.

Lemma 11 *The satisfiability problem for MTL over discrete-time words with variability bounded by v/V is in **EXPSPACE**.*

Proof Given any MTL formula ϕ , build MTL⁺ formula $\phi' = \phi \wedge \mathcal{V}_{v,V}$, where $\mathcal{V}_{v,V}$ is defined in Lemma 9. Then, ϕ' is satisfiable (over generic models) iff ϕ is satisfiable over models with variability bounded by v/V . Since $|\phi'|$ is polynomial in $|\phi|$, $\log v$, and $\log V$, and MTL⁺ is decidable in **EXPSPACE** (Proposition 7), we have proved that the satisfiability problem for MTL over discrete-time words with variability bounded by v/V is in **EXPSPACE**. \square

By combining the previous lemma with the trivial matching lower bound over discrete time we have a tight bound.

Corollary 12 *The satisfiability problem for MTL over discrete-time words with variability bounded by v/V is **EXPSPACE**-complete.*

Complexity in special cases Even if the asymptotic worst-case complexity of MTL over discrete time does not change with bounded variability, our previous work [21, 22] indicates that the complexity decreases in practice for MTL formulas that can be expressed with certain syntactic restrictions, where the exponential blow-up due to succinct encoding of constants is essentially avoided. Therefore, we can still assert that bounded variability may help simplify MTL decidability in some specific cases.

4.3 Bounded variability over dense time

The asymptotic complexity of deciding MTL is significantly lower over models with bounded variability; whereas MTL is normally undecidable over dense time, it is fully decidable and as complex as its discrete-time counterpart if variability is bounded.

We studied the complexity of MTL over models with bounded variability in previous work [18];⁴ in this section we outline how some of those results apply to the present paper's setting.

The lower bound follows the hardness proofs of Theorem 19 in [19].

Lemma 13 *The satisfiability problem for MTL over dense-time words with variability bounded by v/V is **EXPSPACE**-hard.*

⁴Also see the extended version of that work [19], which deals with the interpretations that are applicable to the present paper.

Proof (idea) The proof reduces MTL satisfiability of discrete-time words, which is **EXPSpace**-complete, to MTL satisfiability over dense-time words with bounded variability, by associating discrete instants of time to positions in the word. Bounded variability ensures that there is a maximum number of instants over each time unit, which matches the discrete-time semantics. \square

The upper bound can be derived along the lines of Corollary 12 in [19].

Lemma 14 *The satisfiability problem for MTL over dense-time words with variability bounded by v/V is in **EXPSpace**.*

Proof (idea) Given any MTL formula ϕ , consider MTL⁺ formula $\phi' = \phi \wedge \mathcal{V}_{v,V}$, where $\mathcal{V}_{v,V}$ is defined in Lemma 9. Then, ϕ' is satisfiable (over generic models) iff ϕ is satisfiable over models with variability bounded by v/V .

ϕ' is in general undecidable over dense time since MTL⁺ is a superset of the undecidable MTL. However, it is possible to build a formula ψ in a decidable logic such that, over words with variability bounded by v/V , ψ is satisfiable iff ϕ is. Constructing ψ uses similar techniques as the proof of Corollary 12 in [19]. The basic idea is to use a QITL-like decidable logic [18] to succinctly describe sequences of events over consecutive time intervals of length V . Bounded variability ensures that this can be done by enumeration since there are at most v events over any such interval.

The key point is expressing subformulas such as $\diamond_{=d}(p)$ without using singular intervals, since these are not expressible directly in QITL-like logics or in MITL. For example, if $d < V$ and there are exactly v events within time V , the formula is equivalent to $\Delta = p \vee \bigcirc(p \vee \bigcirc(p \vee \dots))$, with $v - 1$ nested occurrences of \bigcirc .

Then, Δ —and similar formulas for other cases—can be expressed in a QITL-like logic succinctly, that is avoiding the explicit unrolling of v unary operators. This is because QITL-like logics includes operators that can express “counting” properties—such as “ p occurs in one of the next v positions”—with a formula of size logarithmic in v . The same operators can also express $\mathcal{V}_{v,V}$ succinctly as $\mathcal{V}'_{v,V}$, as the negation of “there are $v + 1$ valid positions over the next V time units”. We omit the details, which are somewhat fastidious—because they require dealing with numerous special cases—but straightforward after [18].

In all, we build a formula $\psi' = \psi \wedge \mathcal{V}'_{v,V}$ that has size polynomial in $|\phi|$, $\log v$, and $\log V$; is satisfiable (over finitely-variable dense-time models) iff ϕ is satisfiable over dense-time models with variability bounded by v/V ; and uses a QITL-like logic that is decidable in **EXPSpace** [18]. \square

Corollary 15 *The satisfiability problem for MTL over dense-time words with variability bounded by v/V is **EXPSpace**-complete.*

The following result will also be useful in the rest of the paper.

Corollary 16 *Given integers v and v' , $0 < V \in \mathbb{R}_{\geq 0}$, and an MTL formula ϕ , it is decidable whether there exists a timed word w over $\mathbb{R}_{\geq 0}$ such that w satisfies ϕ , and w has variability bounded by v'/V but not by v/V .*

Proof Consider the formulas $\mathcal{V}_{v,V}$ and $\mathcal{V}_{v',V}$, built according to Lemma 9. The proof of Lemma 14 outlines how to build formulas ψ' , $\mathcal{V}'_{v,V}$, and $\mathcal{V}'_{v',V}$ such that $\mathcal{V}'_{v,V} \wedge \neg \mathcal{V}'_{v',V} \wedge \psi'$ is satisfiable if and only if an w such as in the corollary’s statement exists, which can be decided in **EXPSpace** by Lemma 14. \square

5 Counter machines

Counter machines [38, 58] are powerful computational devices, widely used in formal language theory. Following an approach pioneered by Alur and Henzinger [3, 4] which relates them to MTL, we use counter machines to establish the complexity of deciding bounded variability over dense time. This section introduces a nondeterministic version of counter machines,⁵ and derives some complexity results about them that will be extensively used in Section 6.

Definition 17 An n -counter machine executes programs consisting of a finite list of instructions with locations labeled ℓ_0, ℓ_1, \dots and operating on n integer counter variables v_0, \dots, v_{n-1} . An instruction is one of the following:

| | |
|--|-----------------------|
| halt | terminate computation |
| if $v_k > 0$ goto ℓ_i, ℓ_j | conditional branch |
| inc v_k | increment counter |
| dec v_k | decrement counter |

where the conditional branch consists in jumping to ℓ_i or ℓ_j nondeterministically if counter v_k is non-zero (when $\ell_i = \ell_j$ we only write **goto** ℓ_i — for brevity); and decrementing a counter with value equal to 0 is undefined. Computations start at location ℓ_0 with all counters equal to 0 and proceed according to the obvious semantics of instructions. Without loss of generality, assume that, in every program, instruction **halt** occurs exactly once and the last instruction in the program's list is either **halt** or a branch.

For n -counter machines, with $n \geq 2$, consider the classic problems of halting, non-halting, and recurring computation, which are described in Table 3. The complexity of halting and non-halting is an immediate consequence of the well-known complexity of the same problems for Turing machine computations [57] and the fact that n -counter machines and Turing machines can simulate each other [38, 58]; the complexity of recurring computation is known for 2-counter machines [4], which immediately generalizes to n -counter machines as well. Throughout the paper, we only consider counter machines with at least two counters (the computational power of 1-counter machines is strictly lower, as their halting problem is decidable).

5.1 Bounded and unbounded counters

We introduce three new decision problems for counter machines that involve the value of the counters.

bounded counter: given an integer β , decide whether counter v_0 overflows β in *some* computation;

⁵Our version of counter machines Definition 17 uses nondeterministic jump instructions conditional on testing a counter for being non-zero (greater than zero); while Alur and Henzinger's work [4] uses the more common version that tests a counter for being zero, the two versions are known to be equivalent [58, Appendix A].

Table 3 The complexity of halting, non-halting, and recurring computation for n -counter machines, $n \geq 2$

| PROBLEM | DESCRIPTION | COMPLEXITY |
|-----------------------|---|--------------------------------------|
| halting | Is the location with halt visited in <i>some</i> computation? | $\Sigma_1^0 = \mathbf{RE}$ -complete |
| non-halting | Is the location with halt <i>never</i> visited in <i>some</i> computation? | Σ_2^0 -complete |
| recurring computation | Is location ℓ_0 is visited <i>infinitely often</i> in <i>some</i> computation? | Σ_1^1 -complete |

finite counter: decide whether there exists β such that counter $v_0 \leq \beta$ in *all* computations;
unbounded counter: decide whether counter v_0 is incremented *infinitely often* in *some* computation.

We can prove that the complexity of these decision problems matches that of halting, non-halting, and recurring computation.

Theorem 18 *For n -counter machines with $n \geq 3$ counters: the bounded counter problem is Σ_1^0 -complete; the finite counter problem is Σ_2^0 -complete; the unbounded counter problem is Σ_1^1 -hard.*

Proof We prove hardness by reduction from, respectively, the halting, non-halting, and recurring computation problems of n -counter machines, with $n \geq 2$. Then, we prove completeness by providing matching upper bounds. Since the reductions that prove hardness introduce a fresh counter v_0 with constrained behavior, the theorem holds for machines with at least three counters.

Hardness of the bounded counter problem Given a generic n -counter machine M , we reduce halting to bounded counter for $\beta = 0$ by modifying M into M' as follows. Add one counter and injectively rename all counters in the instruction list so that the new counter is called v_0 ; thus, v_0 is not mentioned in the renamed instructions. Then, replace the unique halting instruction appearing at some ℓ_h in M by two instructions: ℓ_h : **inc** v_0 followed by ℓ_h^+ : **halt**.

Since we only added deterministic instructions, there is a one-to-one correspondence between computations of M and computations of M' . A generic nondeterministic computation χ of M reaches location ℓ_h iff the unique corresponding computation χ' of M' also reaches ℓ_h . In such computations χ' , v_0 overflows β before halting at ℓ_h^+ . In all, some computation of M halts iff v_0 overflows in some computation of M' . Thus, the bounded counter problem is Σ_1^0 -hard.

Hardness of the finite counter problem Given a generic n -counter machine M , we reduce from the non-halting problem.

Consider another counter machine M_d that consists of a deterministic simulation of M . M_d keeps track of the path taken in the computation tree that M 's computations determine, so that it can enumerate M 's computations sequentially. Whenever M_d is simulating

a computation of M that halts, it backtracks the simulation to the most recent unexplored branch and makes a different nondeterministic choice; that is M_d enumerates M 's computations depth-first. Let ℓ_d the location in M_d where the simulation starts over with a new computation—or halts if all computations of M have been exhaustively simulated. The full details of M_d are quite complicated, as they involve keeping track of intermediate computation states (needed) using only the simple memory offered by counter machine; however, it can be done because counter-machines have the same computational power as Turing machines. Furthermore, we can assume that M_d uses the same number of $n \geq 2$ counters as M .

Let us now modify M_d into M' by adding a fresh counter v_0 . Then, add the instruction **inc** v_0 just before ℓ_d ; and replace the unique halting instruction appearing at some ℓ_h in M_d by two instructions in M' : ℓ_h : **inc** v_0 followed by ℓ_h^+ : **if** $v_0 > 0$ **goto** ℓ_h . Let us show that M' has only one non-halting computation. Either (a) M has only finitely many computations, all halting; or (b) M has infinitely many computations, or at least one computation that does not halt. If (a) is the case, after M_d is done simulating all halting computations it halts, whereby M' enters the infinite loop at ℓ_h that makes v_0 diverge. If (b) is the case, M_d 's simulation never halts, and neither does M' , whereby v_0 is incremented once for every halting computation of M .

Consider now the finite counter problem. If the answer to the finite counter problem for M' is YES, it means that (b) is the case (otherwise, M' would enter the infinite loop that makes v_0 diverge), and, furthermore, that v_0 is incremented a finite number of times; therefore, M had at least one non-halting computation (if M had infinitely many halting computations v_0 would diverge). Thus, the answer to the non-halting problem for M is also YES. If the answer to the finite counter problem for M' is NO, it means that v_0 diverges; this can only happen if all computations in M are halting. Thus, the answer to the non-halting problem for M is also NO. In all, the non-halting problem for M has answer YES iff the finite counter problem for M' has answer YES. Thus, the finite counter problem is Σ_2^0 -hard.

Hardness of the unbounded counter problem Given a generic n -counter machine M , we reduce recurring computation to unbounded counter by modifying M into M' as follows. M' has a fresh counter called v_0 . The first instruction of M' is ℓ_0 : **inc** v_0 , followed by M 's program without modifications.

Since we only added one deterministic instruction, there is a one-two-one correspondence between computations of M and computations of M' . A generic nondeterministic computation χ of M visits location ℓ_0 infinitely often iff the unique corresponding computation χ' of M' also reaches the new ℓ_0 infinitely often; such computations χ' increment v_0 infinitely often when executing ℓ_0 . In all, some computation of M visits ℓ_0 infinitely often iff v_0 is incremented infinitely often in some computation of M' . Thus, the unbounded counter problem is Σ_1^1 -hard.

Completeness of the bounded counter problem We reduce the bounded counter problem (for any β) to halting, thus showing that the former is in Σ_1^0 (and hence, by combining it with the hardness result, Σ_1^0 -complete). Given a counter machine M , we modify it into another machine M' as follows. Let ℓ'_h be the new halting location in M' ; instead, M 's halting location ℓ_h becomes an unconditional branch ℓ_h : **goto** ℓ_h that produces an infinite loop in M' ; therefore, computations that halt in M become non-halting computations in M' .

Furthermore, every increment to v_0 is guarded in M' by a conditional:⁶ if $v_0 \geq \beta$ goto ℓ'_h else inc v_0 . Since v_0 is initially zero, a computation of M' halts iff the corresponding computation in M overflowed; hence, a semi-decision procedure for halting produces a semi-decision procedure for bounded counter.

Completeness of the finite counter problem We show that the finite counter problem is in Σ_2^0 (and hence, by combining it with the hardness result, Σ_2^0 -complete) according to the definition of Σ_2^0 in the arithmetical hierarchy [57]. Let \mathcal{O}_β be the set of all counter machines where v_0 overflows β in some computation. We have shown that \mathcal{O}_β is Σ_1^0 for each β ; hence its complement set $\overline{\mathcal{O}_\beta}$ —all counter machines where $v_0 \leq \beta$ in all computations—is Π_1^0 . The set \mathcal{F} of all counter machines for which the finite counter problem has answer YES is defined by $M \in \mathcal{F} \iff \exists \beta : \overline{\mathcal{O}_\beta}$, and hence it is Σ_2^0 .

Completeness of the unbounded counter problem We reduce the unbounded counter problem to the recurring computation problem, thus showing that the former is in Σ_1^1 (and hence, by combining it with the hardness result, Σ_1^1 -complete). Given a counter machine M , we modify it into another machine M' as follows. Add a fresh counter v_j to M' ; as first instruction in M' , add ℓ_0 : **goto** ℓ_{v_j+1} , that is a jump⁷ to the location that corresponds to the $(v_j + 2)$ -th instruction, where v_j is the value of counter v_j ; replace every increment instruction ℓ_k : **inc** v_0 in M by ℓ_k : $v_j := k$ followed by ℓ_{k+1} : **goto** ℓ_0 and then by ℓ_{k+2} : **inc** v_0 —relabeling all other instructions by shifting. The overall behavior of M' is the same as M , except that every time M' increments v_0 it first jumps to location ℓ_0 and back. Since ℓ_0 is visited *only* upon incrementing v_0 , some computation of M increments v_0 infinitely often (unbounded counter) iff some computation of M' visits ℓ_0 infinitely often (recurring computation). □

5.2 MTL and counter machines

Real-time logics such as MTL over $\mathbb{R}_{\geq 0}$ have enough expressive power to encode arbitrary computations of counter machines, as first shown by Alur and Henzinger [3]. We recall their techniques and cast them in our framework, using the continuous semantics of MTL over infinite words.

The idea is that, given an arbitrary n -counter machine M , we can build an MTL formula Γ_M such that there is a one-to-one correspondence between the words that satisfy Γ_M and the computations of M : each word represents the sequence of values taken by counters in the corresponding computation of M .

Consider an n -counter machine M with $m + 1$ instructions ℓ_0, \dots, ℓ_m , such that ℓ_h is the location of the unique halt instruction. A computation of M is entirely described by a sequence c_0, c_1, \dots of configurations, each c_i a tuple $\langle \ell_k, x_0, \dots, x_{n-1} \rangle$ denoting that M is at location ℓ_k , about to execute the corresponding instruction, with the counters storing the values x_0, \dots, x_{n-1} . A computation c_0, c_1, \dots is encoded over a timed word as a sequence of adjacent intervals of unit length: each time interval $[i, i + 1)$ with integer endpoints $i \in \mathbb{N}$ encodes configuration c_i . That is, the propositions holding over $[i, i + 1)$ encode the value of counters and the current location in c_i .

⁶The details of how to encode an if-then-else using counter machine instructions are standard.

⁷Again, the details of how to encode this jump and the other derived instructions using basic counter machine instructions are standard.

In order to encode such information, let us introduce the following propositions: p_k , for $0 \leq k \leq m$, which holds at the left endpoint of unit intervals $[i, i + 1)$ that encode configurations c_i where M is at location ℓ_k ; and z_k , for $0 \leq k < n$, which represents the value of counter v_k as follows: there are as many distinct occurrences of proposition z_k inside a unit interval $(i, i + 1)$ as the value of counter v_k in the corresponding configuration c_i .

To define such an encoding in MTL, let us first constrain what happens at integer times. Exactly one of the propositions p_k 's holds (the current location), with p_0 holding initially at time 0 (the initial location is ℓ_0). Since we want the propositions z_d 's to only occur inside the unit intervals (i.e., not at the endpoints), they are all false at integer times; conversely, the p_k 's are all false at non-integer times. These constraints become:

$$p_0 \wedge \left(\bigwedge_{0 \leq k \leq m} \Box \left(p_k \Rightarrow \bigwedge_{0 \leq j \neq k \leq m} \neg p_j \wedge \bigwedge_{0 \leq d < n} \neg z_d \right) \wedge \right. \\ \left. \bigwedge_{0 \leq k \leq m} \Box \left(p_k \Rightarrow \bigvee_{0 \leq j \leq m} \mathbf{U}_{=1} \left(\bigwedge_{0 \leq i \leq m} \neg p_i, p_j \right) \right) \right).$$

With similar formulas, we constrain the z_k 's to occur at distinct instants: whenever z_k then $\neg z_h$ also holds simultaneously, for all $h \neq k$.

The initial configuration $(\ell_0, 0, \dots, 0)$ is encoded by

$$\bigwedge_{0 \leq j < n} \Box_{[0,1]} (\neg z_j).$$

The encoding of any instruction refers to a current time $t \in \mathbb{N}$ and defines the state over $[t + 1, t + 2)$ as a modification of the state over $[t, t + 1)$. The most significant operation is the increment: ℓ_k : **inc** v_c : the next interval $[t + 1, t + 2)$ includes exactly one more occurrence of z_c than there are in the current interval $[t, t + 1)$:

$$\Box \left(p_k \Rightarrow \left(\begin{array}{l} \Diamond_{=1} p_{k+1} \\ \wedge \bigwedge_{0 \leq d \neq c < n} \Box_{(0,1)} (z_d \Leftrightarrow \Diamond_{=1} z_d) \\ \wedge \Box_{(0,1)} (z_c \Rightarrow \Diamond_{=1} z_c) \\ \wedge \mathbf{U}_{(0,1)} \left(\begin{array}{l} \Diamond_{=1} z_c \Rightarrow z_c, \\ \neg z_c \wedge \Diamond_{=1} z_c \wedge \\ \mathbf{U}_{>0} (\neg z_c \wedge \Diamond_{=1} (\neg z_c), p_{k+1}) \end{array} \right) \end{array} \right) \right). \tag{3}$$

In (3)'s consequent, the first conjunct states that ℓ_{k+1} is the next location visited, since the instruction at ℓ_k is not a branching instruction. The second conjunct states that the values of all counters other than v_c are unchanged: for every occurrence of some z_d in the current interval, there is an occurrence exactly one time unit later in the next interval and vice versa; hence occurrences of z_d are “copied” from the current to the next interval. Similarly, the third conjunct declares that v_c does not decrease (all occurrences of z_c in the current interval are copied into the next one). The fourth conjunct asserts that there exists an instant x in the current interval, after the last occurrence of z_c in the current interval and before the next occurrence of p_{k+1} at the beginning of the next interval, such that z_c occurs exactly once at

$x + 1$ in the next interval; that is z_c holds at $x + 1$ but not at z_c . This new distinct occurrence of z_c is always possible thanks to the density of the temporal domain; thus any value of counters can be stored in a unit time interval. The encoding of other instructions is similar, with the halting instruction determining an indefinite repetition of the final configuration in the future.

Remark 19 The construction shown above that encodes counter-machine computations by means of MTL formulas requires dense time *and* the continuous semantics to work. Specifically, encoding counter increment (3) requires the continuous semantics to be able to match an instant x in the current interval *where no events occur* (hence, x does not correspond to a valid position in the timed word) but where an event occur one time unit later.

Under the pointwise semantics, expressing (3) requires using past operators. The key observation in this case [45] is that the copy of a counter v_d can be expressed as $\Box_{(0,1)}(z_d \Rightarrow \Diamond_{=1} z_d)$ and $\Box_{(1,2)}(z_d \Rightarrow \overleftarrow{\Diamond}_{=1} z_d)$, where $\overleftarrow{\Diamond}_{=1}(\phi)$ holds iff its arguments held one time unit in the past.

6 The complexity of bounded variability

We are ready to prove the main result of the paper: the complexity of deciding whether an MTL formula is satisfied only by models with bounded variability. Section 6.1 rigorously defines two variants of this decision problem; Section 6.2 proves the undecidability of the problems for MTL over dense timed words under the continuous semantics; Section 6.3 proves the complexity of the problems for MTL over discrete timed words under the pointwise semantics; Section 6.4 discusses the complexity of the problems in variants of the two main interpretations.

6.1 Bounded variability problems

The main results of this paper concern the complexity of deciding whether an arbitrary MTL formula is satisfied *only* by words with bounded variability; we now formally define two variants of this decision problem.

Given an interpretation $I = \langle \mathbb{T}, \circ, v/V, \mathbf{s} \rangle$ and an MTL formula ϕ , the decision problems *bounded variability* $\mathbf{BV}_{v/V} \langle \mathbb{T}, \circ, \mathbf{s} \rangle (\phi)$ and *existential bounded variability* $\mathbf{BV}_{\exists} \langle \mathbb{T}, \circ, \mathbf{s} \rangle (\phi)$ are defined as:

Bounded variability $\mathbf{BV}_{v/V} \langle \mathbb{T}, \circ, \mathbf{s} \rangle (\phi)$: Does every model of ϕ under interpretation $\langle \mathbb{T}, \circ, \infty, \mathbf{s} \rangle$ have variability bounded by v/V ? That is, does $\mathcal{B}_{\mathbb{T}}^{\circ}[\infty](\phi)_{\mathbf{s}} \subseteq \mathcal{B}_{\mathbb{T}}^{\circ}[v/V](\phi)_{\mathbf{s}}$?

Existential bounded variability $\mathbf{BV}_{\exists} \langle \mathbb{T}, \circ, \mathbf{s} \rangle (\phi)$: Do there exist $v \in \mathbb{N}$ and $0 < V \in \mathbb{T}$ such that the answer to $\mathbf{BV}_{v/V} \langle \mathbb{T}, \circ, \mathbf{s} \rangle (\phi)$ is YES? That is, does $\exists v, V : \mathcal{B}_{\mathbb{T}}^{\circ}[\infty](\phi)_{\mathbf{s}} \subseteq \mathcal{B}_{\mathbb{T}}^{\circ}[v/V](\phi)_{\mathbf{s}}$?

A bar denotes the corresponding *complement* problems:

$\overline{\mathbf{BV}}_{v/V} \langle \mathbb{T}, \circ, \mathbf{s} \rangle (\phi)$: Does some model of ϕ under interpretation $\langle \mathbb{T}, \circ, \infty, \mathbf{s} \rangle$ have variability not bounded by v/V (that is, bounded by v'/V for some $v' > v$, or unbounded)?

$\overline{\mathbf{BV}}_{\exists} \langle \mathbb{T}, \circ, \mathbf{s} \rangle (\phi)$: Does there exist, for every $v \in \mathbb{N}$ and $0 < V \in \mathbb{T}$, a model of ϕ under interpretation $\langle \mathbb{T}, \circ, \infty, \mathbf{s} \rangle$ that has variability not bounded by v/V ?

Remark 20 Problem $\overline{\mathbf{BV}}_{v/V}(\mathbb{T}, \circ, \mathbf{s})(\phi)$ has answer YES if some model of ϕ under interpretation $\langle \mathbb{T}, \circ, \infty, \mathbf{s} \rangle$ has: (a) variability bounded by v'/V for some $v' > v$; or (b) unbounded variability.

Problem $\overline{\mathbf{BV}}_{\exists}(\mathbb{T}, \circ, \mathbf{s})(\phi)$ has answer YES if: (a) some model of ϕ under interpretation $\langle \mathbb{T}, \circ, \infty, \mathbf{s} \rangle$ has unbounded variability; or (b) every model w of ϕ under interpretation $\langle \mathbb{T}, \circ, \infty, \mathbf{s} \rangle$ has variability bounded by some v_w/V , but the set $\{v_w \mid w \in \mathcal{B}_{\mathbb{T}}^{\circ}[\infty](\phi)_{\mathbf{s}}\}$ does not have a finite upper bound (that is, each model has bounded variability, but no variability bounds all models).

As for MTL formulas, the size of a problem instance is the size of its encoding as a string, which includes a succinct encoding of v and V in the case of the bounded variability problems.

Now that we have defined the two bounded variability problems, we prove their complexity and decidability for different interpretations in the following Table 4 summarizes the results.

6.2 Complexity of bounded variability over dense time

The decision problems $\mathbf{BV}_{v/V}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi)$ and $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi)$ correspond to those of Section 6.1 for MTL interpretations over infinite words ω , dense time domain $\mathbb{R}_{\geq 0}$, and the continuous semantics \mathbf{c} . Both problems are undecidable; their undecidability degrees in the arithmetical hierarchy are different and both lower than MTL satisfiability (which is Σ_1^1 -hard [3], see Section 3.3).

Theorem 21 $\mathbf{BV}_{v/V}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi)$ is $\Pi_1^0 = \mathbf{coRE}$ -complete; $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi)$ is Σ_2^0 -complete.

We split the proof of Theorem 21 in four lemmas, which separately establish lower and upper bounds for each problem.

Lemma 22 $\mathbf{BV}_{v/V}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi)$ is in $\Pi_1^0 = \mathbf{coRE}$.

Proof We give a procedure to semi-decide $\overline{\mathbf{BV}}_{v/V}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi)$; this establishes that $\overline{\mathbf{BV}}_{v/V}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi) \in \mathbf{RE}$ and thus $\mathbf{BV}_{v/V}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi) \in \mathbf{coRE}$ by complement.

Table 4 Summary of the main results of this paper: complexity of deciding bounded variability and existential bounded variability of MTL formulas under different interpretations. For each choice of time domain \mathbb{T} , finite (*) or infinite (ω) timed words, and pointwise or continuous semantics, the table shows whether BOUNDED VARIABILITY $\mathbf{BV}_{v/V}(\mathbb{T}, \circ, \mathbf{s})(\phi)$ and EXISTENTIAL BOUNDED VARIABILITY $\mathbf{BV}_{\exists}(\mathbb{T}, \circ, \mathbf{s})(\phi)$ is DECidable (yes or no) under that interpretation and the complexity of the problem

| \mathbb{T} | \circ | \mathbf{s} | BOUNDED VAR. | | EXISTENTIAL BOUNDED VAR. | |
|----------------------------------|------------|-------------------------|--------------|-------------------------------------|--------------------------|------------------------|
| | | | DEC? | COMPLEXITY | DEC? | COMPLEXITY |
| $\mathbb{N}/\mathbb{R}_{\geq 0}$ | $*/\omega$ | \mathbf{p}/\mathbf{c} | | | | |
| \mathbb{N} | | | Y | EXPSpace -complete | Y | trivial |
| $\mathbb{R}_{\geq 0}$ | * | \mathbf{p} | Y | non-primitive recursive | | RE |
| $\mathbb{R}_{\geq 0}$ | ω | \mathbf{p} | N | $\Pi_1^0 = \mathbf{coRE}$ -complete | | Σ_2^0 |
| $\mathbb{R}_{\geq 0}$ | | \mathbf{c} | N | $\Pi_1^0 = \mathbf{coRE}$ -complete | N | Σ_2^0 -complete |

As pointed out in Theorem 20, some model of ϕ has variability not bounded by v/V iff: (a) it has variability bounded by v'/V but not by v/V , for some $v' > v$; or (b) it has unbounded variability. Since timestamps are diverging (according to the definition of timed words in Section 2), (b) can only occur if, for any finite time t , the variability is bounded by some v_t , but such v_t diverges as t goes to infinity.⁸

For any finite time T , let $\phi[T]$ denote an MTL formula that restricts the evaluation of ϕ to the finite time interval $[0, T]$. This can be constructed as follows: add a fresh proposition e constrained by $\phi_e = \bigcup_{=T}(e, e \wedge \square_{>0} \neg e)$. Rewrite ϕ in negation normal form, and replace every atom q by $e \Rightarrow q$. Postulate that, if e is false, all other propositions in \mathcal{P} are false as well: $\phi_{\mathcal{P}} = \square(\neg e \Rightarrow \bigwedge_{p \in \mathcal{P}} \neg p)$. Finally, $\phi[T]$ is $\phi_e \wedge \phi \wedge \phi_{\mathcal{P}}$. Since no event occurs after finite time T , all models of $\phi[T]$ have variability bounded by x/T , for some finite (possibly very large) x . Therefore, the only way in which $\phi[T]$ can have some model with variability not bounded by some x'/T is if $x' < x$.

We can now describe a procedure P_1 that semi-decides $\overline{\mathbf{BV}}_{v/V}(\mathbb{R}_{\geq 0}, \omega, c)(\phi)$; it consists of the following steps:

1. Initially, $\delta := v + 1$ and $\Delta := V + 1$;
2. Using Corollary 16, decide whether $\phi[\Delta]$ has some model with variability bounded by δ/V but not by v/V ;
3. If it does, stop and return YES;
4. Otherwise $\delta := \delta + 1$, $\Delta := \Delta + 1$, and go to (2).

If the answer to $\overline{\mathbf{BV}}_{v/V}(\mathbb{R}_{\geq 0}, \omega, c)(\phi)$ is YES, then either (a) or (b) above holds; let us show that, in both cases, P_1 terminates with the correct answer.

If (a) is the case, let w_a be a model with variability bounded by v'/V but not by v/V for some $v' > v$; that is, w_a has \bar{v} events, for $v < \bar{v} \leq v'$, over some time interval $[y, y + V]$. In this case, P_1 terminates with YES as soon as $\delta \geq \bar{v}$ and $\Delta \geq y + V$.

If (b) is the case, let w_b be a model with unbounded variability; since variability is unbounded, there exists a time T such that: w_b has $v' > v$ events over some time window $[y, y + V]$, for $0 \leq y < y + V \leq T$. In this case, P_1 terminates with YES as soon as $\delta \geq v'$ and $\Delta \geq T$. □

Lemma 23 $\overline{\mathbf{BV}}_{v/V}(\mathbb{R}_{\geq 0}, \omega, c)(\phi)$ is **coRE-hard**.

Proof We reduce the bounded counter problem of n -counter machines, $n \geq 3$, (Section 5.1) to $\overline{\mathbf{BV}}_{v/V}(\mathbb{R}_{\geq 0}, \omega, c)(\phi)$; the lemma follows by Theorem 18 through complement problems.

Consider a generic n -counter machine M with counters v_0, \dots, v_{n-1} . We construct an MTL formula Γ_M that encodes the computations of M along the lines of Section 5.2, but with some modifications. For $t \in \mathbb{N}$, the t -th configuration $\langle \ell_k, x_0, \dots, x_{n-1} \rangle$ is encoded over the time interval $[2nt, 2nt + 2n)$ as follows: p_k holds at $2nt$; for $k = 0, \dots, n - 1$, z_k holds x_k times over $(2nt + 2k + 1, 2nt + 2k + 2)$; and no propositions hold elsewhere over the whole $[2nt, 2nt + 2n)$. With this spacing of counter events, the models of Γ_M are such that any interval of length 1 includes at most as many events as the largest value held by a counter during some computation of M . Thus, Γ_M has some model with variability

⁸In related work, we called similar behaviors ‘‘Berkeley’’ [17, 20].

not bounded by $\beta/1$ —problem $\overline{\mathbf{BV}}_{\beta/1}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\Gamma_M)$ —iff a counter overflows β in some computation of M .

Now we have only established whether *some* counter overflows in M , whereas the bounded counter problem specifically targets overflows of v_0 . To close the gap, we encode the overflowing of v_0 in M as an MTL formula:

$$\Xi_{\beta}^{v_0} = \diamond \left(\left(\bigvee_{0 \leq k \leq m} p_k \right) \wedge \overbrace{\diamond_{(0,1)} (z_0 \wedge \diamond_{(0,1)} (z_0 \wedge \dots))}^{\beta+1 \text{ nested diamonds}} \right).$$

Thanks to the padding, the nested diamonds in $\Xi_{\beta}^{v_0}$ evaluate to true iff there are at least $\beta + 1$ distinct occurrences of z_0 in the slot corresponding to one configuration. Thus, v_0 overflows β in M iff $\Gamma_M \wedge \Xi_{\beta}^{v_0}$ has some model with variability not bounded by $\beta/1$. \square

Lemma 24 $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi)$ is in Σ_2^0 .

Proof Given the definition of Σ_2^0 in the arithmetical hierarchy [57], it is sufficient to provide an enumeration of all MTL formulas ϕ for which the answer to $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi)$ is YES, relative to an oracle for $\mathbf{BV}_{v/V}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi)$, which is in Π_1^0 by Lemma 22. To this end, we dovetail [48, Chap. 3] through all pairs (v, ϕ) of nonnegative integers $v \in \mathbb{N}$ and MTL formulas ϕ . For each pair, if the answer to $\mathbf{BV}_{v/1}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi)$ is YES, then the answer to $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi)$ also is YES. This enumeration eventually finds all formulas ϕ for which the answer to $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi)$ is YES. \square

Lemma 25 $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi)$ is Σ_2^0 -hard.

Proof We reduce the finite counter problem of counter machines (Section 5.1) to $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi)$; the lemma follows by Theorem 18.

Let M be a generic n -counter machine M , with $n \geq 3$. Consider another counter machine M_b that performs a deterministic simulation of M . M_b keeps track of the path taken in the computation tree that M 's computations determine, so that it can enumerate M 's computations sequentially. Unlike M_d in the hardness proof of the finite counter problem (Theorem 18), M_b uses dovetailing [48, Chap. 3] to enumerate M 's computation breadth-first: whenever M_b reaches an unexplored branch in the simulation of a computation of M , it performs one step into one unexplored branch, and then backtracks and performs one step into the alternative choice. We further require that M_b does not directly use counter v_0 for the simulation (and adds fresh counters as needed); instead, it enforces the invariant that v_0 always stores the current value of v_0 in the computation that is being simulated.

Next, modify M_b into M' as follows. The unique halting location ℓ_h in M_b becomes an infinite idle loop in M' : ℓ_h : **goto** ℓ_h . Add a fresh counter v_x ; before beginning the main computation, M' initializes v_x with a nondeterministically chosen positive integer value x by performing the instructions:

$$\begin{aligned} \ell_0 &: \mathbf{inc} \ v_x \\ \ell_1 &: \mathbf{if} \ v_x > 0 \ \mathbf{goto} \ \ell_0, \ell_2 \end{aligned} \tag{4}$$

where ℓ_2 corresponds to M_b 's initial location after shifting all location labels by two positions. Furthermore, every increment **inc** v_0 of v_0 in M_b becomes a guarded increment **if** $v_0 = v_x - 1$ **goto** ℓ_s else **inc** v_0 in M' , where ℓ_s : **halt** is a fresh halting location only used by these guarded increments. Except for the nondeterministic initialization of v_x , M' only adds deterministic instructions to the deterministic M_b ; hence, for every positive value x that may initialize v_x , there exists a computation of M' where $v_x = x$; let c_x denote such a computation of M' that initializes v_x to x and then proceeds with the deterministic simulation (where v_x remains constant to x). Computation c_x halts iff there exists a computation of M where v_0 overflows $x - 1$.

Let $\Gamma = \Gamma_{M'} \wedge \diamond p_s$, where $\Gamma_{M'}$ is an MTL formula that encodes the computations of M' as described in Section 5.2, and ℓ_s is the unique halting location of M' . The models of Γ describe all valid computations of M' that halt; that is, in particular, all computations that do not get stuck forever in the initial loop (4) that increments v_x .⁹

Finally consider the decision problem $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\Gamma)$. If its answer is YES, we claim that there are finitely many models that satisfy Γ .¹⁰ To the contrary, assume that Γ has infinitely many models; then, a counting argument indicates that, for every positive y , there exists $x > y$ such that $w_x \models \Gamma$, where w_x is a model of the halting computation c_x . But v_0 overflows $x - 1$, and hence $y \leq x - 1$ too, in c_x ; this contradicts the hypothesis that the answer to $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\Gamma)$ is YES (i.e., there is a finite bound on v_0 's values shared by all computations of M).

Conversely, if the answer to $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\Gamma)$ is NO, then there are infinitely many models that satisfy Γ , which implies, using a dual argument as the YES case, that the finite counter problem for M has answer NO. □

Remark 26 As a comment on the proof of Lemma 25, its reduction's trickiest part lied in the fact that, while the finite counter problem refers a specific counter v_0 , $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi)$ considers variability of all propositions. The proof of Lemma 23 involves a similar mismatch, but things are simpler there thanks to the existence of a known bound β , which can be monitored explicitly (formula $\Xi_{\beta}^{v_0}$); in the finite variability problem, instead, the bound is existentially quantified. An obvious solution would have been to avoid the problem altogether by changing the definition of $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi)$ so that it referred to a specific proposition whose variability should be bounded; doing this, however, would provide a weaker result about a definition that is possibly less natural. Instead, we leveraged nondeterminism to "guess" the existentially quantified bound x .

6.3 Complexity of bounded variability over discrete time

The decision problems $\mathbf{BV}_{v/V}(\mathbb{N}, \omega, \mathbf{p})(\phi)$ and $\mathbf{BV}_{\exists}(\mathbb{N}, \omega, \mathbf{p})(\phi)$ correspond to those of Section 6.1 for MTL interpretations over infinite words ω , discrete time domain \mathbb{N} , and the pointwise semantics \mathbf{p} . Both problems are decidable; the complexity of the former is the same as MTL satisfiability under the same interpretation, whereas the latter is trivial given that the time domain is discrete.

⁹We can specify and reason about progress because we are reducing between undecidable problems.

¹⁰In the special case in which M has finitely many halting computations, Γ has no models.

Theorem 27 $\mathbf{BV}_{v/V}(\mathbb{N}, \omega, \rho)(\phi)$ is **EXSPACE**-complete; $\mathbf{BV}_{\exists}(\mathbb{N}, \omega, \rho)(\phi)$ is trivially decidable (constant answer YES).

Proof For any integer $x > 0$, every word $w \in \mathcal{B}_{\mathbb{N}}^{\omega}[\infty]$ has variability trivially bounded by $(x + 1)/x$ because the time domain is discrete and the timestamps are monotonically increasing, and hence the difference between a timestamp t_k and its next t_{k+1} is at least one unit. This shows that $\mathbf{BV}_{\exists}(\mathbb{N}, \omega, \rho)(\phi)$ has answer YES for every ϕ .

The first part of the theorem is proved in Lemma 28 and 29. □

Lemma 28 $\mathbf{BV}_{v/V}(\mathbb{N}, \omega, \rho)(\phi)$ is **EXSPACE**-hard.

Proof It suffices to note that $\mathbf{BV}_{2/1}(\mathbb{N}, \omega, \rho)(\phi)$ has answer YES iff ϕ is satisfiable over infinite words with monotonic integer timestamps. This provides a constant-time reduction from an **EXSPACE**-complete problem to $\mathbf{BV}_{v/V}(\mathbb{N}, \omega, \rho)(\phi)$. □

Lemma 29 $\mathbf{BV}_{v/V}(\mathbb{N}, \omega, \rho)(\phi)$ is in **EXSPACE**.

Proof To prove the lemma, we follow Lemma 9 and express bounded variability in the MTL^+ extension of MTL. Given an MTL formula ϕ , the MTL^+ formula $\phi' = \phi \Rightarrow \mathcal{V}_{v,V}$ is valid iff $\mathbf{BV}_{v/V}(\mathbb{N}, \omega, \rho)(\phi)$ has answer YES. This proves the lemma since MTL^+ validity is in **EXSPACE**. □

6.4 Complexity of bounded variability under other interpretations

In the previous sections, we studied the complexity of bounded variability and existential bounded variability for certain standard choices of interpretation: infinite words, the pointwise semantics in discrete time, and the continuous semantics in dense time. We now discuss to what extent those results generalize to other interpretations; Table 4 summarizes the results.

6.4.1 Discrete time

In Section 3.3, we remarked how the decidability of MTL is very robust over discrete time, in that complexity is not affected by choices of interpretation such as pointwise vs. continuous semantics and finite vs. infinite words.

We observe a similar robustness regarding the complexity of the bounded variability and existential bounded variability problems over discrete time \mathbb{N} . It should be clear that the proofs of Section 6.3 can be adapted to work for the other discrete-time interpretations with minimal modifications. Accommodating finite-word interpretations requires a bit of attention to the “border effects” that occur when reaching the last positions in a word; accommodating the continuous semantics requires handling the case where there is a valid position k for every nonnegative integer timestamp $t_k = k$, similarly to what is done when bridging the gap between MTL and LTL (see Section 7.1 for some details of how this simple idea works).

Theorem 30 $\mathbf{BV}_{v/V}(\mathbb{N}, \omega, c)(\phi)$, $\mathbf{BV}_{v/V}(\mathbb{N}, *, \rho)(\phi)$, and $\mathbf{BV}_{v/V}(\mathbb{N}, *, c)(\phi)$ are **EXSPACE**-complete; $\mathbf{BV}_{\exists}(\mathbb{N}, \omega, c)(\phi)$, $\mathbf{BV}_{\exists}(\mathbb{N}, *, \rho)(\phi)$, and $\mathbf{BV}_{\exists}(\mathbb{N}, *, c)(\phi)$ are trivially decidable (constant answer YES).

6.4.2 Dense time

As recalled in Table 2, the complexity of MTL changes significantly under different dense-time interpretations. This affects the complexity of bounded variability and existential bounded variability over dense time.

Continuous semantics The results of Section 6.2 still hold over finite words under the continuous semantics: the proofs of Lemmas 22, 23, 24, and 25 go through also for finite-word interpretations. The intuitive reason is that the counter machine problems we reduce from (the bounded counter and finite counter problems) only involve finite behaviors: overflow of a counter is an event that occurs at some finite time, that is bounded counter and finite counter are safety properties [32, 37]. This is a sufficient condition for the lower bounds to hold for finite words as well (Lemmas 23 and 25); the upper bounds follow a fortiori since finite words can be seen as a restriction of infinite words.

Theorem 31 $\mathbf{BV}_{v/V}(\mathbb{R}_{\geq 0}, *, c)(\phi)$ is $\Pi_1^0 = \mathbf{coRE}$ -complete; $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, *, c)(\phi)$ is Σ_2^0 -complete.

Pointwise semantics: bounded variability. Let us now consider interpretations over dense time with the pointwise semantics. Notice that Lemma 9 provides a way of showing that, for every dense-time interpretation $I = \langle \mathbb{R}_{\geq 0}, o, v/V, s \rangle$, the complexity of the bounded variability problem $\mathbf{BV}_{v/V}(\mathbb{R}_{\geq 0}, o, s)(\phi)$ is not higher than the complexity of validity for MTL under the same interpretation I . In fact, given an MTL formula ϕ , $\mathbf{BV}_{v/V}(\mathbb{R}_{\geq 0}, o, s)(\phi)$ has answer YES iff the MTL formula $\phi \Rightarrow \mathcal{V}_{v,V}$ is valid under interpretation I , where $\mathcal{V}_{v,V}$ is described in Lemma 9. In particular, the bounded variability problem is non-primitive recursive for finite words ($\mathbf{BV}_{v/V}(\mathbb{R}_{\geq 0}, *, p)(\phi)$) and in **coRE** for infinite words ($\mathbf{BV}_{v/V}(\mathbb{R}_{\geq 0}, \omega, p)(\phi) \in \mathbf{coRE}$).

To establish matching lower bounds, we rely on a property S_δ of MTL over the pointwise interpretation: a formula ϕ is satisfiable if and only if it is satisfied by a word w such that, for every valid position k in w , $t_{k+1} - t_k \leq \delta$, where δ is a constant that depends only on ϕ . This property depends on the abstraction of clock valuations into clock regions [45]: roughly speaking, the number of regions that can be reached in a computation corresponding to a word satisfying ϕ is finite, which implies that, if ϕ is satisfiable, one can construct a word w that satisfies ϕ with time differences bounded by some finite δ . Therefore, we can reduce satisfiability to $\overline{\mathbf{BV}}_{1/\delta}(\mathbb{R}_{\geq 0}, o, p)(\phi)$, for both $o = *$ and $o = \omega$. If $\overline{\mathbf{BV}}_{1/\delta}(\mathbb{R}_{\geq 0}, o, p)(\phi)$ has answer NO, then every word w that satisfies ϕ is such that, for all valid positions k in w , $t_{k+1} - t_k > \delta$; therefore, the above property S_δ does not hold, and hence ϕ is not satisfiable. If $\overline{\mathbf{BV}}_{1/\delta}(\mathbb{R}_{\geq 0}, o, p)(\phi)$ has answer YES, then there exists a word w that satisfies ϕ and such that, for some valid position k in w , $t_{k+1} - t_k \leq \delta$; therefore ϕ is satisfiable a fortiori.

Theorem 32 $\mathbf{BV}_{v/V}(\mathbb{R}_{\geq 0}, *, p)(\phi)$ is complete for non-primitive recursive problems; $\mathbf{BV}_{v/V}(\mathbb{R}_{\geq 0}, \omega, p)(\phi)$ is **coRE**-complete.¹¹

Pointwise semantics: existential bounded variability. It is easy to show that the existential bounded variability problems $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, *, p)(\phi)$ and $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, \omega, p)(\phi)$ are one

¹¹The preliminary version of this paper [23, Sec. VII] is incorrect when it suggests that $\mathbf{BV}_{v/V}(\mathbb{R}_{\geq 0}, \omega, p)(\phi)$ is complete for **RE**.

level up in the arithmetical hierarchy with respect to the corresponding bounded variability problems.

For the finite word interpretation, let us describe an enumeration of formulas ϕ for which the answer to problem $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, *, \rho)(\phi)$ is YES. This proves that the problem is in **RE**. Here is the enumeration: (1) initially, $\delta := 1$; (2) check whether $\phi \Rightarrow \mathcal{V}_{\delta,1}$ (where $\mathcal{V}_{\delta,1}$ encodes the bounded variability requirement as per Lemma 9) is valid; (3) if it is, stop and return YES; (4) otherwise, $\delta := \delta + 1$ and go to (2). Setting $V = 1$ is without loss of generality, because any dense-time word that has variability bounded by v/V also has variability bounded by $(v(1 + \lfloor 1/V \rfloor))/1$.

Theorem 33 $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, *, \rho)(\phi)$ is in **RE**.

For the infinite word interpretation, we apply the definition of Σ_2^0 in the arithmetical hierarchy [57]. We have just shown that the set $\mathcal{O}_{v,V}$ of all MTL formulas ϕ such that problem $\mathbf{BV}_{v/V}(\mathbb{R}_{\geq 0}, \omega, \rho)(\phi)$ has answer YES is **coRE** (or Π_1^0). Therefore the set $\{\phi \mid \exists v, V : \phi \in \mathcal{O}_{v,V}\}$ is Σ_2^0 by definition.

Theorem 34 $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, \omega, \rho)(\phi)$ is in Σ_2^0 .¹²

Finding matching lower bounds for $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, *, \rho)(\phi)$ and $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, \omega, \rho)(\phi)$ requires novel techniques, which belongs to future work. We remark that one cannot use reductions from n -counter machine problems, because the correspondence between counter-machine computations and MTL formulas breaks down under the pointwise semantics (see Remark 19). Instead, one could try to exploit the connection between MTL under the pointwise semantics and channel machines with insertion errors [44, 45].

Past operators. Remark 19 outlined how dense-time MTL *with past operators* can encode counter-machine computations even under the pointwise semantics. This implies that the constructions for continuous-semantics MTL used in the proofs of Lemmas 22–25 work, *mutatis mutandis*, for pointwise-semantics MTL with past operators. More precisely, the proofs of Lemmas 22 and 24—which establish upper bounds—do not involve counter-machines, and hence carry over to the pointwise semantics without modifications. The proofs of Lemmas 23 and 25—which establish lower bounds—depend on being able to construct an MTL formula Γ_M that encodes the computations of an arbitrary counter machine M ; if we replace the encoding of Γ_M described in Section 5.2—using continuous-semantics MTL—with the encoding outlined in Remark 19—using pointwise-semantics MTL with past operators—the proofs carry over to the pointwise semantics as well. In all, the complexities of the bounded variability problem and of the existential bounded variability problem for dense-time MTL with past operators under the pointwise semantics are the same as for dense-time MTL under the continuous semantics.

¹²The preliminary version of this paper [23, Sec. VII] is incorrect when it suggests that $\mathbf{BV}_{\exists}(\mathbb{R}_{\geq 0}, \omega, \rho)(\phi)$ is in **RE**.

7 Bounded variability in simple cases

The complexity results of Section 6 pose some major limitations to deciding bounded variability for MTL formulas. However, there is still the possibility of identifying syntactic *fragments* of MTL that are still reasonably expressive but for which reasoning about bounded variability is simpler than in the general case. We refer to such MTL fragments as “bounded friendly”.

The idea is that a bounded friendly MTL fragment \mathcal{F} is one for which the problem $\mathbf{BV}_{v/V}(\mathbb{T}, \circ, \mathbf{s})(\psi)$, for $\psi \in \mathcal{F}$, is simpler than the more general problem $\mathbf{BV}_{v/V}(\mathbb{T}, \circ, \mathbf{s})(\phi)$, for $\phi \in \text{MTL}$. If we are studying the satisfiability of an MTL formula ϕ that can be written as $\phi' \wedge \psi$, for $\psi \in \mathcal{F}$, we can proceed as follows. First, we determine if $\mathbf{BV}_{v/V}(\mathbb{T}, \circ, \mathbf{s})(\psi)$ has answer YES. If it does, then ϕ has variability bounded by v/V a fortiori; hence, we can use the simpler algorithms of Section 4 to study the satisfiability of ϕ . This approach is best effort, in that ϕ may not be expressible as $\phi' \wedge \psi$ or ψ may not have variability bounded by v/V , but it has a chance of leading to an overall simpler decision procedure in practical cases.

We give two examples of non-trivial bounded-friendly fragments, one for discrete and one for dense time.

7.1 Simpler bounded variability over discrete time

As recalled in Section 4.2, MTL over discrete time boils down to an exponentially succinct version of LTL. Therefore, we can lift some complexity results about simpler fragments of LTL [5, 12, 59] to MTL over \mathbb{N} , and use them to identify bounded-friendly fragments. As customary over discrete time, we consider interpretations with infinite words and the pointwise semantics, although extending the results to the case of finite words is possible (and not particularly interesting).

To this end, we have to recall a few fundamental definitions of LTL and explicitly connect them to MTL. In keeping with the standard LTL models, we assume that words have one valid position per integer time: $t_k = k$ for all $k \in \mathbb{N}$; this is merely a representational convention that does not affect the properties of MTL over discrete time, since one can always convert a generic word into one with one valid position per integer time by adding a silent padding event ϵ at all positions corresponding to timestamps that did not appear in the generic word.

Linear temporal logic (LTL). Linear Temporal Logic (LTL) formulas ψ are recursively defined by:

$$\psi ::= \top \mid p \mid \neg\psi_1 \mid \psi_1 \wedge \psi_2 \mid \psi_1 \bar{\mathbf{U}} \psi_2 \mid \bar{\mathbf{X}} \psi_1,$$

where the qualitative until and next operator’s semantics is given in MTL by:

| SYMBOL | DEFINITION | NAME |
|----------------------------------|--|-------------------|
| $\psi_1 \bar{\mathbf{U}} \psi_2$ | $\psi_2 \vee \left(\psi_1 \wedge \mathbf{U}_{[0,\infty)}(\psi_1, \psi_2) \right)$ | qualitative until |
| $\bar{\mathbf{X}} \psi_1$ | $\bigcirc_{=1}(\psi_1)$ | LTL next |

Two dual MTL fragments. Consider the two dual MTL fragments $\mathcal{M}_{\diamond, \circ}^+$ and $\mathcal{M}_{\square, \circ}^+$: $\mathcal{M}_{\diamond, \circ}^+$ (respectively, $\mathcal{M}_{\square, \circ}^+$) denotes the MTL fragment using only the \diamond_J (respectively, \square_J) and \circ_J temporal operators (which we now regard as primitive), the propositional connectives \wedge and \vee , and where negations only appear on atomic propositions. We can prove that satisfiability for these fragments is decidable in exponential *time*.

Lemma 35 *Satisfiability of $\mathcal{M}_{\diamond, \circ}^+$ and of $\mathcal{M}_{\square, \circ}^+$ over \mathbb{N} is EXP-complete.*

Proof Consider the LTL fragment $\mathcal{L}_{\bar{F}, \bar{X}}^+$ which only uses the LTL operators eventually \bar{F} and next \bar{X} , the propositional connectives \wedge and \vee , and where negations only appear on atomic propositions. The LTL operator \bar{F} is defined as $\bar{F} \psi_1 = \top \bar{U} \psi_1$. A classic result is that satisfiability for $\mathcal{L}_{\bar{F}, \bar{X}}^+$ is NP-complete [59, Th. 3.7].

We now outline how to transform a generic $\mu \in \mathcal{M}_{\diamond, \circ}^+$ into a $\lambda \in \mathcal{L}_{\bar{F}, \bar{X}}^+$ such that μ and λ are equisatisfiable; the converse transformation (from $\mathcal{L}_{\bar{F}, \bar{X}}^+$ to $\mathcal{M}_{\diamond, \circ}^+$ formulas) is trivial given that MTL is a superset of LTL the way we have defined it. In general, the size of λ will be exponential in the size of μ due to the fact that metric constraints are encoded in binary in μ . The lemma follows as a manifestation of the “succinctness phenomenon” [48, Chap. 20]—from NP to EXP.

Recall that we assume, without loss of generality over discrete time, words where each valid position k has a timestamp t_k equal to k itself; this way, a step always corresponds to one discrete time instant. A special proposition ϵ holds when no significant event takes place. A translation τ from $\mathcal{M}_{\diamond, \circ}^+$ to $\mathcal{L}_{\bar{F}, \bar{X}}^+$ works inductively as follows, for $a, b \in \mathbb{N}$, $c \in \mathbb{N} \cup \{\infty\}$, and $p \in \mathcal{P}$:

$$\begin{aligned} \tau \left(\diamond_{[a,b]}(p) \right) &= \overline{X}^a \left(p \vee \overbrace{\bar{X} \left(p \vee \dots \right)}^{b-a \text{ nested } \bar{X}} \right), \\ \tau \left(\diamond_{[a,\infty)}(p) \right) &= \overline{X}^a \bar{F} (p), \\ \tau \left(\circ_{[a,c]}(p) \right) &= \overbrace{\bar{X} \left(\epsilon \wedge \bar{X} \left(\epsilon \wedge \dots \right) \right)}^{a-1 \text{ nested } \bar{X}} \wedge \tau \left(\diamond_{[a,c]}(p) \right), \end{aligned}$$

where \bar{X}^k is a shorthand for k nested applications of \bar{X} . Translation τ does not otherwise change the propositional structure of formulas.

With this translation, it should be clear that μ is satisfiable over timed words over \mathbb{N} iff $\tau(\mu)$ is satisfiable. The size of $\tau(\mu)$ is $O(2^{|\mu|})$, since τ unrolls the constants, succinctly represented in μ , which results in worst-case exponential blow-up. This establishes the lemma for $\mathcal{M}_{\diamond, \circ}^+$.

The same complexity result for $\mathcal{M}_{\square, \circ}^+$ follows by duality of \square and \diamond . □

We can leverage Lemma 35 to show that $\mathcal{M}_{\diamond, \circ}^+$ is *bounded friendly*.¹³ Let v, V be variability bounds; without loss of generality, let $v > 0$. First, note that the $\mathcal{M}_{\square, \circ}^+$ formula $B_{v, V} = \square \square_{(0, v]}(\perp)$,¹⁴ where $v = \lceil V/v \rceil$, holds only for models with variability bounded

¹³By duality, one could show that $\mathcal{M}_{\diamond, \circ}^+$ is bounded friendly too.

¹⁴Or, equivalently, $\square \circ_{> v}(\top)$.

by v/V (specifically, it is stricter than the requirement of bounded variability). Consider now a generic MTL formula ϕ written as $\phi' \wedge \psi$, where $\psi \in \mathcal{M}_{\diamond, \circ}^+$. The implication $\psi \Rightarrow B_{v,V} \equiv \neg\psi \vee B_{v,V}$ is an $\mathcal{M}_{\square, \circ}^+$ formula: push in the outermost negation $\neg\psi$, and use the duality between \diamond and \square . Thus, validity of $\psi \Rightarrow B_{v,V}$ can be decided in singly exponential time (Lemma 35), which is better than deciding $\mathbf{BV}_{v/V}(\mathbb{N}, \omega, \rho)(\phi)$ or the validity of ϕ for general models (both are **EXPSpace**-complete problems, and hence solving them takes *time* doubly exponential in $|\phi|$ in the worst case).

7.2 Simpler bounded variability over dense time

While full MTL is highly undecidable over dense time, a number of expressive yet decidable fragments thereof have been identified. MITL is the fragment of MTL where intervals that decorate time operators are not singular; MITL is fully decidable with **EXPSpace**-complete complexity [2, 28]. More recently, other decidable expressive fragments have been identified that allow singular intervals [46] in MTL; BMTL and SMTL, in particular, are interesting because their expressive power is incomparable with MITL’s. As in the previous section, we consider interpretations with infinite words and the continuous semantics.

From the point of view of identifying bounded friendly fragments of MTL, however, MITL seems the most promising choice. SMTL validity is non-elementary; while this is still better than the undecidable problem $\mathbf{BV}_{v/V}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi)$, it remains impractical. BMTL validity, in contrast, is decidable in **EXPSpace**; however, BMTL cannot express invariance properties since only finite intervals are allowed, and it is clear that bounded variability is a form of invariance since it has to hold over every position in a timed word.

We are left with MITL, which we can show to be bounded friendly. Let v, V be variability bounds; without loss of generality, let $v > 0$. First, note that the MITL formula $B_{v,V} = \square\square_{(0,v]}(\perp)$, where $v = \lceil V/v \rceil$, holds only for models with variability bounded by v/V (specifically, it is stricter than then requirement of bounded variability). Consider now a generic MTL formula ϕ written as $\phi' \wedge \psi$, where $\psi \in \text{MITL}$. The implication $\psi \Rightarrow B_{v,V}$ is obviously also an MITL formula. Thus, validity of $\psi \Rightarrow B_{v,V}$ can be decided in **EXPSpace**, whereas deciding both $\mathbf{BV}_{v/V}(\mathbb{R}_{\geq 0}, \omega, \mathbf{c})(\phi)$ and the validity of ϕ is undecidable in general. If $\psi \Rightarrow B_{v,V}$ is valid, ϕ has bounded variability a fortiori, and hence its validity can be decided in **EXPSpace** too.

8 Conclusion

The original motivation for this paper’s study was leveraging bounded variability to simplify reasoning with metric temporal logic. From this perspective, the results we presented are a mixed bag. Over discrete time models, deciding whether an MTL formula has intrinsic bounded variability is not simpler than deciding validity. Over dense time models, deciding whether an MTL formula has intrinsic bounded variability generally has complexity lower than deciding validity, but unfortunately remains undecidable or intractable. This means that checking for intrinsic bounded variability first, and then applying the simpler algorithms for validity, is not feasible in practice. Nevertheless, we can still expect that some constraints that determine bounded variability may occur naturally as a consequence of the physical requirements of real systems being modeled. Specifically, bounded variability seems to naturally embody a notion of *bounded speed* of transition between states, to which most physical processes would comply.

From the different perspective of scientific interest, there remain a number of open problems consisting in transposing some of the techniques and results of the paper to other semantics and logics. Finding matching lower bound in some uncommon semantic interpretations of MTL over dense time (Section 6.4) would require using novel techniques involving channel machines with insertion errors [44, 45]. Other, broader and hence potentially more interesting, open questions involve studying bounded variability for different kinds of metric temporal logics, including recent proposals for classic [17, 55, 56] or interval-based [8, 39] real-time notations.

Acknowledgments We thank the editors for organizing this special issue; the attendees and reviewers of TIME 2014 for their suggestions about a preliminary version of this work; and the anonymous referees of AMAI for their scrupulous work and valuable comments.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Abadi, M., Lamport, L.: An old-fashioned recipe for real-time. *ACM Trans. Program. Lang. Syst.* **16**(5), 1543–1571 (1994)
2. Alur, R., Feder, T., Henzinger, T.A.: The benefits of relaxing punctuality. *J. ACM* **43**(1), 116–146 (1996)
3. Alur, R., Henzinger, T.A.: Real-time logics: Complexity and expressiveness. *Inf. Comp.* **104**(1), 35–77 (1993)
4. Alur, R., Henzinger, T.A.: A really temporal logic. *J. ACM* **41**(1), 181–204 (1994)
5. Bauland, M., Schneider, T., Schnoor, H., Schnoor, I., Vollmer, H.: The complexity of generalized satisfiability for linear temporal logic. *Logical Methods in Computer Science* **5**(1) (2009)
6. Bouyer, P., Markey, N., Ouaknine, J., Worrell, J.: The Cost of Punctuality. In: *ACM/IEEE Symposium on Logic in Computer Science*, pp. 109–120 (2007)
7. Bresolin, D., Della Monica, D., Goranko, V., Montanari, A., Sciavicco, G.: The dark side of interval temporal logic: Sharpening the undecidability border. In: *International Symposium on Temporal Representation and Reasoning*, pp. 131–138 (2011)
8. Bresolin, D., Della monica, D., Goranko, V., Montanari, A., Sciavicco, G.: Metric propositional neighborhood logics on natural numbers. *Softw. Syst. Model.* **12**(2), 245–264 (2013)
9. Bresolin, D., Monica, D.D., Montanari, A., Sala, P., Sciavicco, G.: Interval temporal logics over strongly discrete linear orders: Expressiveness and complexity. *Theor. Comput. Sci.* **560**, 269–291 (2014)
10. Bresolin, D., Montanari, A., Sala, P., Sciavicco, G.: Optimal decision procedures for MPNL over finite structures, the natural numbers, and the integers. *Theor. Comput. Sci.* **493**, 98–115 (2013)
11. Chaochen, Z., Hansen, M.R., Sestoft, P.: Decidability and Undecidability Results for Duration Calculus. In: Enjalbert, P., Finkel, A., Wagner, K.W. (eds.) *STACS 93, 10th Annual Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science*, vol. 665, pp. 58–68. Springer (1993)
12. Demri, S., Schnoebelen, P.: The complexity of propositional linear temporal logics in simple cases. *Inf. Comput.* **174**(1), 84–103 (2002)
13. D’Souza, D., Prabhakar, P.: On the expressiveness of MTL in the pointwise and continuous semantics. *STTT* **9**(1), 1–4 (2007)
14. D’Souza, D., Prabhakar, P.: On the expressiveness of MTL in the pointwise and continuous semantics. *STTT* **9**(1), 1–4 (2007)
15. Emerson, E.A.: Temporal and Modal Logic. In: *Handbook of Theoretical Computer Science*, vol. B, pp. 996–1072. Elsevier Science (1990)
16. Fränzle, M.: Model-checking dense-time duration calculus. *Formal Asp. Comput.* **16**(2), 121–139 (2004)
17. Furia, C.A., Mandrioli, D., Morzenti, A., Rossi, M.: *Modeling Time in Computing. Monographs in Theoretical Computer Science. An EATCS series* Springer (2012)
18. Furia, C.A., Rossi, M.: MTL with bounded variability: Decidability and complexity. In: *FORMATS, LNCS*, vol. 5215, pp. 109–123. Springer (2008). Extended version in [19]

19. Furia, C.A., Rossi, M.: MTL with bounded variability: Decidability and complexity. Tech. Rep. 2008.10, Dipartimento di Elettronica e Informazione, Politecnico di Milano. Available at <http://bugcounting.net/publications.html#MTLwBoundedVar-TR08> (2008)
20. Furia, C.A., Rossi, M.: A theory of sampling for continuous-time metric temporal logic. *ACM Transactions on Computational Logic* **12**(1), 1–40 (2010). Article 8
21. Furia, C.A., Spoletini, P.: On Relaxing Metric Information in Linear Temporal Logic. In: *International Symposium on Temporal Representation and Reasoning*, pp. 72–79. IEEE (2011)
22. Furia, C.A., Spoletini, P.: Automata-Based Verification of Linear Temporal Logic Models with Bounded Variability. In: *International Symposium on Temporal Representation and Reasoning*, pp. 89–96. IEEE (2012)
23. Furia, C.A., Spoletini, P.: Bounded Variability of Metric Temporal Logic. In: Cesta, A., Combi, C., Laroussinie, F. (eds.) *Proceedings of the 21st International Symposium on Temporal Representation and Reasoning (TIME'14)*, pp. 155–163. IEEE Computer Society (2014)
24. Gabbay, D.M., Hodkinson, I., Reynolds, M.: *Temporal Logic (vol. 1): mathematical foundations and computational aspects*, Oxford Logic Guides, vol. 28. Oxford University Press (1994)
25. Gabbay, D.M., Pnueli, A., Shelah, S., Stavi, J.: On the Temporal Basis of Fairness. In: *Conference Record of the 7Th Annual ACM Symposium on Principles of Programming Languages (POPL'80)*, pp. 163–173 (1980)
26. Hirshfeld, Y., Rabinovich, A.: Logics for real time: Decidability and complexity. *Fundam. Inf.* **62**(1), 1–28 (2004)
27. Hirshfeld, Y., Rabinovich, A.: Continuous time temporal logic with counting. *Inf. Comput.* **214**, 1–9 (2012)
28. Hirshfeld, Y., Rabinovich, A.M.: Logics for real time: Decidability and complexity. *Fundam. Inform.* **62**(1), 1–28 (2004)
29. Hunter, P., Ouaknine, J., Worrell, J.: Expressive Completeness for Metric Temporal Logic. In: *LICS*, pp. 349–357. IEEE (2013)
30. Kamp, J.A.W.: *Tense Logic and the Theory of Linear Order*. Ph.D. Thesis. University of California, Los Angeles (1968)
31. Koymans, R.: Specifying real-time properties with metric temporal logic. *Real-Time Systems* **2**(4), 255–299 (1990)
32. Lamport, L.: Proving the correctness of multiprocess programs. *IEEE Trans. Softw. Eng.* **SE-3**(2), 125–143 (1977)
33. Lutz, C., Walther, D., Wolter, F.: Quantitative temporal logics over the reals: PSPACE and below. *Inf. Comput.* **205**(1), 99–123 (2007)
34. Maler, O., Nickovic, D., Pnueli, A.: Real Time Temporal Logic: Past, present, future. In: Petterson, P., Yi, W. (eds.) *Proceedings of the 3rd International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS'05)*, Lecture Notes in Computer Science, vol. 3829, pp. 2–16. Springer-Verlag (2005)
35. Maler, O., Nickovic, D., Pnueli, A.: From MITL to Timed Automata. In: Asarin, E., Bouyer, P. (eds.) *Proceedings of the 4th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS'06)*, Lecture Notes in Computer Science, vol. 4202, pp. 274–289. Springer-Verlag (2006)
36. Maler, O., Nickovic, D., Pnueli, A.: Checking Temporal Properties of Discrete, Timed and Continuous Behaviors. In: *Pillars of Computer Science, Essays Dedicated to Boris (Boaz) Trakhtenbrot on the Occasion of His 85Th Birthday*, Lecture Notes in Computer Science, vol. 4800, pp. 475–505. Springer (2008)
37. Manna, Z., Pnueli, A.: A Hierarchy of Temporal Properties. In: *Proceedings of the 9Th Annual ACM Symposium on Principles of Distributed Computing*, pp. 377–410. ACM (1990)
38. Minsky, M.L.: *Computation: Finite and infinite machines* prentice hall (1967)
39. Montanari, A., Pazzaglia, M., Sala, P.: Metric Propositional Neighborhood Logic with an Equivalence Relation. In: *21St International Symposium on Temporal Representation and Reasoning, (TIME)*, pp. 49–58. IEEE Computer Society (2014)
40. Montanari, A., Puppis, G., Sala, P.: Decidability of the Interval Temporal Logic $\mathbb{A}\mathbb{A}\mathbb{B}\mathbb{B}$ over the Rationals. In: *Mathematical Foundations of Computer Science 2014 - 39Th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part I*, pp. 451–463 (2014)
41. Montanari, A., Sala, P.: An Optimal Tableau System for the Logic of Temporal Neighborhood over the Reals. In: *19Th International Symposium on Temporal Representation and Reasoning, TIME 2012, Leicester, United Kingdom, September 12-14, 2012*, pp. 39–46 (2012)
42. Nickovic, D., Piterman, N.: From MTL to Deterministic Timed Automata. In: Chatterjee, K., Henzinger, T.A. (eds.) *Formal Modeling and Analysis of Timed Systems – 8th International Conference, FORMATS 2010, Lecture Notes in Computer Science, vol. 6246*, pp. 152–167. Springer (2010)

43. Ouaknine, J., Rabinovich, A., Worrell, J.: Time-Bounded Verification. In: Bravetti, M., Zavattaro, G. (eds.) CONCUR 2009 – Concurrency Theory, 20th International Conference, Lecture Notes in Computer Science, vol. 5710, pp. 496–510. Springer (2009)
44. Ouaknine, J., Worrell, J.: On Metric Temporal Logic and Faulty Turing Machines. In: FoSSaCS, LNCS, vol. 3921, pp. 217–230. Springer (2006)
45. Ouaknine, J., Worrell, J.: On the decidability and complexity of metric temporal logic over finite words. *Logical Methods in Computer Science* **3**(1) (2007)
46. Ouaknine, J., Worrell, J.: Some Recent Results in Metric Temporal Logic. In: FORMATS, LNCS, vol. 5215, pp. 1–13. Springer (2008)
47. Ouaknine, J., Worrell, J.: Towards a Theory of Time-Bounded Verification. In: Abramsky, S., Gavoille, C., Kirchner, C., auf der Heide, F.M., Spirakis, P.G. (eds.) Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Lecture Notes in Computer Science, vol. 6199, pp. 22–37. Springer (2010)
48. Papadimitriou, C.: Computational complexity Addison-Wesley (1994)
49. Perrin, D., Pin, J.É.: Infinite Words, Pure and Applied Mathematics, vol. 141. Elsevier (2004)
50. Pnueli, A.: The Temporal Logic of Programs. In: Proceedings of the 18Th Annual Symposium on Foundations of Computer Science, SFCS '77, pp. 46–57. IEEE Computer Society (1977)
51. Rabinovich, A.: Complexity of Metric Temporal Logics with Counting and the Pnueli Modalities. In: FORMATS, Lecture Notes in Computer Science, vol. 5215, pp. 93–108. Springer (2008)
52. Rabinovich, A.: Complexity of metric temporal logics with counting and the Pnueli modalities. *Theor. Comput. Sci.* **411**(22–24), 2331–2342 (2010)
53. Rabinovich, A.M.: Expressive completeness of Duration Calculus. *Inf. Comput.* **156**(1–2), 320–344 (2000)
54. Reynolds, M.: The complexity of temporal logic over the reals. *Ann. Pure Appl. Logic* **161**(8), 1063–1096 (2010)
55. Reynolds, M.: Metric temporal reasoning with less than two clocks. *Journal of Applied Non-Classical Logics* **20**(4), 437–455 (2010)
56. Reynolds, M.: A New Metric Temporal Logic for Hybrid Systems. In: 20Th International Symposium on Temporal Representation and Reasoning (TIME), pp. 73–80. IEEE Computer Society (2013)
57. Rogers, Jr., H.: Theory of recursive functions and effective computability MIT press (1987)
58. Shepherdson, J.C., Sturgis, H.E.: Computability of recursive functions. *J. ACM* **10**(2) (1963)
59. Sistla, A.P., Clarke, E.M.: The complexity of propositional linear temporal logics. *J. ACM* **32**(3), 733–749 (1985)
60. Vardi, M.Y.: An Automata-Theoretic Approach to Linear Temporal Logic. In: Logics for Concurrency – Structure versus Automata (8Th Banff Higher Order Workshop), Lecture Notes in Computer Science, vol. 1043, pp. 238–266. Springer (1995)
61. Vardi, M.Y., Wolper, P.: An Automata-Theoretic Approach to Automatic Program Verification. In: LICS, pp. 332–344. IEEE (1986)
62. Wilke, T.: Specifying Timed State Sequences in Powerful Decidable Logics and Timed Automata. In: FTRTFT, LNCS, vol. 863, pp. 694–715. Springer (1994)