CHARACTERISING SJT REDUCIBILITY

NOAM GREENBERG, ANDRÉ NIES AND DANIEL TURETSKY

Abstract. SJT reducibility between sets $A, B \subseteq \mathbb{N}$ is defined by $A \leq_{SJT} B$ if for each computable function h that is unbounded and nondecreasing, there is an h-bounded uniformly B-c.e. trace $(T_n)_{n\in\mathbb{N}}$ such that for each n, the value $J^A(n)$ of the jump is in T_n, if defined. This reducibility is slightly weaker than Turing reducibility. We study SJT reducibility, and as a main result give several characterisations of it on the K-trivial sets. This is the first case of extending the three lowness paradigms, weak as an oracle, computed by many, and inert, to the setting of weak reducibilities.

§1. Introduction. The concept of a weak reducibility on sets of natural numbers arises by combining two general notions developed in computability theory from the year 2002 onwards: lowness properties of sets [26, 29] and partial relativisation [2]. A *lowness property* formalises a particular aspect of computational simplicity of a set of natural numbers (henceforth simply called a set). Three paradigms have been proposed [15]: being weak as an oracle, computed by many oracles, and inert (see Section 1.2 below). A *partial relativisation* of a computability theoretic concept $\mathcal{C}$ is obtained by relativising only certain constituents of the concept to an oracle, while the remaining constituents, typically bounds on sizes of finite sets or on the number of changes of a computable approximation of a set, are left unrelativised. See [2, Section 2.1] for examples. It is customary to use the term "$\mathcal{C}$ by X" for a partial relativisation of $\mathcal{C}$ to an oracle X; the constituents of the concept that are relativised are assumed to be understood from the context. Each of these two general notions was obtained by crystalising mathematical intuition that developed through series of works; the citations above merely point to some early places (known to us) where the notions were formulated.

1.1. Weak reducibilities. A pre-ordering $\leq_W$ on sets is called a *weak reducibility* if $\leq_W$ is arithmetical, $X \leq_{\mathrm{T}} Y$ implies $X \leq_W Y$, and $X' \not\leq_W X$ for each X. This terminology [28, Section 5.6] is chosen to be opposite to the terminology of strong reducibilities of Odifreddi [33] (such as truth table reducibility).

Given an arithmetical lowness property $\mathcal{L}$ and a proper choice of its partial relativisation, the binary relation "A is in $\mathcal{L}$ by B" tends to be a weak reducibility (in particular, it is transitive, and implied by $\leq_{\mathrm{T}}$). We consider two examples. The first is the much studied LR-reducibility [26]. Let MLR^X be the class of Martin-Löf random sets relative to X, and recall that A is low for ML-random if $\mathsf{MLR} \subseteq \mathsf{MLR}^A$. LR-reducibility $\leq_{\mathrm{LR}}$ is obtained by partially relativising this to B: for $A \leq_{\mathrm{LR}} B$

Received February 27, 2026 UTC.
2020 *Mathematics Subject Classification.* Primary 03D25, 03D32.
Key words and phrases. reducibility, strong jump traceability, cost function, Demuth randomness.

© The Author(s), 2026. Published by Cambridge University Press on behalf of The Association for Symbolic Logic.
0022-4812/00/0000-0000
DOI:10.1017/jsl.2026.10228

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

to hold, one requires that $\mathsf{MLR}^B \subseteq \mathsf{MLR}^A$, rather than the full relativisation to B, which would be $\mathsf{MLR}^B \subseteq \mathsf{MLR}^{A\oplus B}$.

As a second example, more relevant to the present article, let us consider the lowness property of being jump traceable. Recall that an *order function* is an unbounded, nondecreasing computable function on $\mathbb{N}$. A *c.e. trace* is a uniformly c.e. sequence $\langle T_n\rangle_{n\in\mathbb{N}}$ of finite sets. For an order function h, such a trace is h-bounded if $|T_n| \leq h(n)$ for each n. A trace is computably bounded if it is h-bounded for some computable order function h.

Definition 1.1 [27]. A set A is *jump-traceable* if there is a computably bounded c.e. trace $\langle T_n\rangle_{n\in\mathbb{N}}$ such that for all n, $J^A(n)$ is in T_n if it is defined.

Here J^A is a universal A-partial computable function. It is easy to see that each jump traceable set is GL_1 [28, 8.4.3]. The appropriate partial relativisation to an oracle B, namely A is jump traceable by B, asks that the trace be c.e. in B, but still only J^A, rather than $J^{A\oplus B}$, is traced, and the order function bounding the trace is still computable. This yields the reducibility $\leq_{\mathrm{JT}}$, introduced in [28, 8.4.13], with transitivity verified in 8.4.14. To check that $\leq_{\mathrm{JT}}$ is a weak reducibility, note that it is clearly Σ^0_4, and implied by $\leq_{\mathrm{T}}$. Relativising the fact that each jump traceable is GL_1 one shows that $X' \not\leq_{\mathrm{JT}} X$ for each set X.

The class of jump traceable sets has a perfect Π^0_1 subclass. It admits superhigh members [22], so it cannot be considered a strong lowness property. If one requires that J^A has an h-bounded trace for *every* order function h, one obtains the notion of strong jump traceability, introduced in [12]. This notion has been studied extensively [6, 8, 9, 14, 15, 18, 19, 25], with applications outside randomness [10] (see [20] for a survey). The central notion for this article is a corresponding weak reducibility.

Definition 1.2 (SJT reducibility, [28, 8.4.37]). For sets A, B, one writes $A \leq_{\mathrm{SJT}} B$ if for every order function h, there is a B-c.e., h-bounded trace $\langle T_n\rangle_{n\in\mathbb{N}}$ such that for each n, if $J^A(n)$ is defined then it is in T_n.

Note that a set A is strongly jump traceable iff $A \leq_{\mathrm{SJT}} \emptyset$. An argument similar to [25, Theorem 3.3] or [28, 8.4.14] shows that the relation $\leq_{\mathrm{SJT}}$ is transitive. Clearly $X' \not\leq_{\mathrm{SJT}} X$ for each set X, because this already holds for $\leq_{\mathrm{JT}}$. The relation $\leq_{\mathrm{SJT}}$ is Π^0_5. We also note that $\leq_{\mathrm{SJT}}$ is genuinely weaker than $\leq_{\mathrm{T}}$, in the sense that for each set B there is a set $C >_T B$ such that $C \leq_{\mathrm{SJT}} B$. To see this, by relativisation of a result in [12] let A be a set such that $A \not\leq_{\mathrm{T}} B$ and A is strongly jump traceable relative to B, and let $C = A \oplus B$. Then for each order function h relative to B, J^C has a B-c.e. trace bounded by h.

Remark 1.3. A weak reducibility $\leq_W$ induces the highness property of $\leq_W$-hardness, namely the property of a set A that $\emptyset' \leq_W A$. This highness property is in a sense dual to the lowness property that $\leq_W$ is based on. Such highness properties have often been in the focus of research. Results often state coincidences between highness properties; for instance, a set is LR-hard iff it is uniformly almost everywhere dominating [21]. The highness property of SJT-hardness was first studied in [10], where the authors showed as their main result that some incomputable, c.e. set is Turing below each c.e. SJT-hard set.

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

1.2. Three lowness paradigms. As mentioned, three lowness paradigms for a set $A \subseteq \mathbb{N}$ have been proposed. We give a brief summary (see [15] for more details).

(1) *Weak as an oracle*:
A is not very useful as an oracle for Turing machines.

(2) *Computed by many oracles*:
The collection of oracles computing A is large. The class of such oracles is null unless A is computable, so largeness must be taken in a more specific sense. For instance, A is called a base for ML-randomness if there is an oracle in MLR^A computing A.

(3) *Inert*:
Recall that the Shoenfield limit lemma states that a set A is Δ^0_2 iff it can be computably approximated with a finite number of changes. Inertness means that there is such an approximation where the number of changes is in a sense small.

It is often interesting to ask whether a given lowness property has equivalent definitions according to two or even all three paradigms. We examine this for the property that a set A is low for ML-randomness: every ML-random is ML-random relative to A.

(1) The definition follows Paradigm (1). Another equivalent characterisation according to this paradigm is lowness for K: there is a constant d such that $K(x) \leq K^A(x) + d$ for each string x, where K denotes prefix-free descriptive string complexity.

(2) A is low for ML-randomness iff it is a base for ML-randomness, which gives a definition according to Paradigm (2) (see [28, Chapter 5] for details and references).

(3) A is low for ML-randomness iff it has a computable approximation with few changes in the sense of the cost function $\mathbf{c}_\Omega(x, s) = \Omega_s - \Omega_x$ by [32]; this follows Paradigm (3). An earlier version of this result for a cost function $\mathbf{c}_K$ dates back to [26].

We note that lowness for ML-randomness also coincides with K-triviality, a property expressing that the set is far from random [11, 26].

Strong jump traceability by its definition follows Paradigm (1). In [8, 15] an equivalent characterisation following Paradigm (2) is given: a set is strongly jump traceable if and only if it is computed by every ω-c.a. (or every superlow, or every superhigh) ML-random set.[1] A related result appears in [17]: a set is strongly jump-traceable if and only if it is computable from every infinite "section" of Chaitin's Ω. In [18] it is shown that a c.e. set is strongly jump traceable iff it obeys all benign cost functions; this yields a characterisation in terms of Paradigm (3).

[1]Recall that a function $f : \mathbb{N} \to \mathbb{N}$ is *ω-computably approximable*, or ω-c.a., if it has a computable approximation $f(x, s)$ such that $n \mapsto \#\{s : f(x, s) \neq f(x, s+1)\}$ is bounded by some computable function; equivalently, $f \leq_{\mathrm{wtt}} \emptyset'$. A set $X \subseteq \mathbb{N}$ is ω-c.a. if its characteristic function is ω-c.a. The terminology "ω-c.a." was introduced for clarity, since the term "ω-c.e." is often used for the class Σ^{-1}_ω in Ershov's effective difference hierarchy, which is a larger class of sets (the ω-c.a. sets are those that are both ω-c.e. and co- ω-c.e.)

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

1.3. Characterising SJT-reducibility according to the three paradigms. In this article, we suggest to apply the three paradigms not only to lowness properties, but also to the corresponding weak reducibilities. Our main results show that, under certain restrictions, the characterisations of the strongly jump traceable sets according to the three paradigms can be lifted to the case of the weak reducibility $\leq_{\mathrm{SJT}}$.

The definition of $\leq_{\mathrm{SJT}}$ follows paradigm (1), as does the definition of strong jump-traceability itself. It indicates computational weakness of the oracle A compared to B, because $J^A(n)$ is caught in an "arbitrarily small" trace set enumerated by B. Our first result gives a characterisation according to Paradigm (3): for every benign cost function $\mathbf{c}$, the set B computes an approximation of A for which the total cost of changes in the sense of $\mathbf{c}$ is finite. For the precise definitions of the notation $A \models_B \mathbf{c}$ and of benign cost functions, see Definitions 2.1 and 2.2 below.

THEOREM 1.4. *Suppose that both A and B are jump-traceable. The following are equivalent*:

(a) $A \leq_{SJT} B$.
(b) $A \models_B \mathbf{c}$ *for every benign cost function* $\mathbf{c}$.

We prove Theorem 1.4 in Section 2.

Recall that $A \subseteq \mathbb{N}$ is K-trivial if $K(A\upharpoonright_n) \leq^+ K(n)$ for each n. There are strongly jump traceable sets outside the computable [12], yet the class lies far inside the K-trivials [9, 16]. Recent work [16, 17] reveals a rich structure inside the K-trivial Turing degrees, based on both cost functions and computability from various random sets. This, and the existing relationship between strong jump-traceability and K-triviality, leads us to expect that the weak reducibility $\leq_{\mathrm{SJT}}$ will also shed new light on the K-trivial sets and degrees. Our second result gives a variety of characterisations of $\leq_{\mathrm{SJT}}$ within the K-trivial sets, according to Paradigm (2). We quickly recall the definition of some other classes that will be required.

- A set Y is superhigh if $\emptyset'' \leq_{\mathrm{tt}} Y'$.
- A set Y is ω-c.a. if its characteristic function is ω-c.a.; equivalently if $Y \leq_{\mathrm{wtt}} \emptyset'$.
- A set Y is superlow if $Y' \leq_{\mathrm{tt}} \emptyset'$; equivalently, if Y' is ω-c.a.
- Given an infinite set $R \subseteq \mathbb{N}$, let Ω_R be the result of removing from Ω all the bits in locations not in R. As mentioned above, here Ω is any left-c.e. ML-random set.

THEOREM 1.5. *Let $\mathcal{C}$ be any of the following classes of sets*:

(1) *sets that are not weakly Demuth random*;
(2) *superhigh sets*;
(3) *ω-c.a. sets*;
(4) *superlow sets*;
(5) *the sets Ω_R, where R is infinite and computable.*

Then the following are equivalent for K-trivial sets A and B:

(a) $A \leq_{SJT} B$.
(b) $A \leq_T B \oplus Y$ *for every Martin-Löf-random set* $Y \in \mathcal{C}$.

Note that every K-trivial set is jump traceable. This was first shown by Nies [26], improving a result of Zambella (see [35]). For weak Demuth randomness, see, for example, [3]; we give a detailed definition in Section 3 (Definition 3.2).

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

1.4. The difficulties with partial relativisation. As the restrictions in the two theorems above indicate, lifting the various characterisations of strong jump traceability to the weak reducibility $\leq_{\mathrm{SJT}}$ is far from straightforward. Since arguments in computability theory almost always relativise, the difficulty lies in the fact that $\leq_{\mathrm{SJT}}$ is a partial relativisation.

One of the difficulties is in modifying *adaptive* arguments. In such an argument, we typically work with a strongly jump-traceable oracle A, and define an A-partial computable function ψ. By the recursion theorem, during the construction we have access to a c.e. trace $\langle T_n \rangle_{n\in\mathbb{N}}$ for ψ, and we use that trace to dynamically define ψ: the definition of ψ *adapts* to the enumerations appearing in $\langle T_n \rangle_{n\in\mathbb{N}}$. When $B \neq \emptyset$, more specifically, when A does not compute B, the enumeration of $\langle T_n \rangle_{n\in\mathbb{N}}$ is only B-computable, and so A does not have such access to the trace. In a full relativisation to B we use $A \oplus B$ rather than A, which allows us to use adaptive arguments.

Similar difficulties occur in the study of other weak reducibilities such as $\leq_{\mathrm{LR}}$ mentioned above. However, our characterisation in Theorem 1.4 will show that this problem does not occur with $\leq_{\mathrm{SJT}}$ (under the same assumption on $A \oplus B$).

THEOREM 1.6. *Suppose that $A \oplus B$ is jump-traceable. The following are equivalent*:

(a) $A \leq_{SJT} B$.
(b) $A \oplus B \leq_{SJT} B$.

We discuss Theorem 1.6, and other corollaries of our characterisations, in Section 4. Note that since Turing reducibility implies $\leq_{\mathrm{SJT}}$, $A \oplus B \leq_{\mathrm{SJT}} B$ if and only if $A \oplus B \equiv_{\mathrm{SJT}} B$. Regarding $\leq_{\mathrm{SJT}}$ and joins, see Section 4, in particular Question 4.3 and the discussion thereafter.

§2. Benign cost functions. We recall the definitions of cost functions and approximations obeying them; for background see [28, Chapter 5] or [32].

DEFINITION 2.1. A *cost function* is a computable function $\mathbf{c}\colon \mathbb{N}\times\mathbb{N} \to \mathbb{Q}^+$ satisfying:

- *Monotonicity*: $\mathbf{c}$ is nonincreasing in the first variable and nondecreasing in the second.
- *The limit condition*: For all x, $\lim_s \mathbf{c}(x,s)$ exists, and $\lim_x \lim_s \mathbf{c}(x,s) = 0$.

Let $\langle A_s \rangle$ be an approximation of a set A. The *total* $\mathbf{c}$*-cost* of the approximation is the sum

$$\sum_s \mathbf{c}(x_s, s),$$

where $x_s = |A_s \wedge A_{s+1}|$ is the least x such that $A_s(x) \neq A_{s+1}(x)$.

Let A and B be sets. We write $A \models_B \mathbf{c}$ if there is a B-computable approximation $\langle A_s \rangle_{s\in\mathbb{N}}$ of A for which the total $\mathbf{c}$-cost is finite.

One views $\mathbf{c}(x,s)$ as the cost of changing an approximation at x as the least element, at a stage s. The limit condition says that for large enough x, changing $A(x)$ at any stage is as cheap as one likes.

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

DEFINITION 2.2 [18]. A cost function $\mathbf{c}$ is *benign* if given a rational $\varepsilon > 0$, we can compute a bound on the length of any sequence $x_1 < s_1 \leq x_2 < s_2 \leq \cdots \leq x_\ell < s_\ell$ such that $\mathbf{c}(x_i, s_i) \geq \varepsilon$ for all $i \leq \ell$.

For example, the cost function $\mathbf{c}_\Omega(x, s) = \Omega_s - \Omega_x$ is benign, with the bound for ε being $1/\varepsilon$. Here (Ω_s) is an increasing computable approximation of the left-c.e., ML-random sequence Ω.

2.1. A proof of Theorem 1.6. We first explain how Theorem 1.6 follows from Theorem 1.4. In one direction, suppose that $A \oplus B \leq_{\mathrm{SJT}} B$; since $A \leq_{\mathrm{T}} A \oplus B$ and $\leq_{\mathrm{T}}$ implies $\leq_{\mathrm{SJT}}$, we have $A \leq_{\mathrm{SJT}} B$.

In the other direction, suppose that $A \oplus B$ is jump-traceable and that $A \leq_{\mathrm{SJT}} B$. The assumption on $A \oplus B$ implies that both A and B are jump-traceable, so Theorem 1.4 applies. Hence, $A \models_B \mathbf{c}$ for every benign cost function $\mathbf{c}$. This implies that $A \oplus B \models_B \mathbf{c}$ for every benign cost function $\mathbf{c}$: fix such a cost function $\mathbf{c}$. Suppose that (A_s) is a B-computable approximation of A witnessing that $A \models_B \mathbf{c}$. Let $C_s = A_s \oplus B$. The fact that $\mathbf{c}$ is monotonic implies that the total $\mathbf{c}$-cost of (C_s) is bounded by the total $\mathbf{c}$-cost of (A_s), and so is finite.

The assumption on $A \oplus B$, together with the other direction of Theorem 1.4, now shows that $A \oplus B \leq_{\mathrm{SJT}} B$.

In the next two sections we give a proof of Theorem 1.4.

2.2. Proof of (b) $\Rightarrow$(a) of Theorem 1.4. In this direction, in fact, we do not make use of the assumption that B is jump-traceable. Proposition 2.1 of [18] gives this direction for the case $B = \emptyset$, under the extra assumption that A is c.e. That proof does not partially relativise; however, we can now present a simpler argument that applies more generally.

Suppose that A is jump-traceable, and that $A \models_B \mathbf{c}$ for every benign cost function $\mathbf{c}$. Let h be an order function; we show that J^A has a B-c.e. trace bounded by h. We define an A-partial computable function ψ by setting $\psi(n) = \sigma$ if $J^A(n)\downarrow$ with use $\sigma \prec A$. Since A is jump-traceable, the partial function ψ has a c.e. trace $\langle V_n \rangle_{n \in \mathbb{N}}$, bounded by some computable order function g.

We define a cost function $\mathbf{c}$ as follows:

$$\mathbf{c}(x, s) = \max \{1/h(n) \,:\, \exists \sigma \in V_{n,s} \text{ s.t. } |\sigma| > x\}.$$

The function $\mathbf{c}$ is monotonic: the fact that $V_{n,s} \subseteq V_{n,t}$ when $s < t$ implies that it is increasing in the second variable; it is clearly decreasing in the first variable. It satisfies the limit condition because h is unbounded, and each V_n is finite. We show that $\mathbf{c}$ is benign. Let $\varepsilon > 0$. Suppose that $s < y \leq t$ and that $\mathbf{c}(y, t) \geq \varepsilon$. There is some n with $1/h(n) \geq \varepsilon$ and some $\sigma \in V_{n,t}$ with $|\sigma| > y$. Since $|\sigma| > s$ we may assume that $\sigma \notin V_{m,s}$ for any m. Hence, the length of a sequence $x_1 < s_1 \leq x_2 < s_2 \leq x_3 < s_3 \leq \cdots$ with $\mathbf{c}(x_i, s_i) \geq \varepsilon$ is bounded by the size of

$$\bigcup \{V_n \,:\, 1/h(n) \geq \varepsilon\},$$

which is bounded by

$$\sum \{g(n) \,:\, 1/h(n) \geq \varepsilon\}.$$

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

Since h is computable, non-decreasing, and unbounded, and g is computable, this sum is finite and can be obtained computably from ε.

Now suppose that (A_s) is a B-computable approximation whose total $\mathbf{c}$-cost is finite; by ignoring finitely many stages of the approximation, we may assume that the total cost is bounded by 1. For each n, let T_n be the set of values $J_s^{A_s}(n)$ for s such that $J_s^{A_s}(n)\downarrow$ with use $\sigma \prec A_s$, and $\sigma \in V_{n,s}$. If $J^A(n)\downarrow$ then since $\psi(n) \in V_n$, we have $J^A(n) \in T_n$. Let $s < t$ be two stages responsible for enumerating new values into T_n; let $\sigma \prec A_s$ with $J_s^{A_s}(n) = J^\sigma(n)$ and $\sigma \in V_{n,s}$. By definition, $\mathbf{c}(x, s) \geq 1/h(n)$ for every $x < |\sigma|$. Since $J_t^{A_t}(n) \neq J_s^{A_s}(n)$, we have $\sigma \not\prec A_t$, so there is some stage r with $s \leq r < t$ and some $x < |\sigma|$ such that $A_r(x) \neq A_{r+1}(x)$. Thus, the contribution of stage r to the total $\mathbf{c}$-cost of the approximation (A_s) is at least $1/h(n)$. This shows that $|T_n| \leq h(n) + 1$. Replacing h by $h - 1$ gives the desired result.

2.3. Proof of (a) $\Rightarrow$ (b) of Theorem 1.4. We turn to the proof of the harder direction of Theorem 1.4. The case $B = \emptyset$ is Theorem 1.12 of [8]. That argument is adaptive; we overcome this by using the jump-traceability of B. Very roughly, we define an A-partial computable function Φ^A, which will have a B-c.e. trace $\langle U_x^B \rangle_{x\in\mathbb{N}}$. Instead of directly defining values of Φ^A in reaction to elements being enumerated into the trace, we use the fact that B is jump-traceable to "trace the trace"; we will have a c.e. trace $\langle V_y \rangle_{y\in\mathbb{N}}$, with no oracle, that essentially traces values in $\langle U_x^B \rangle_{x\in\mathbb{N}}$, and we define Φ^A in reaction to values showing up in $\langle V_y \rangle_{y\in\mathbb{N}}$. The thing to note is that the size of $\langle U_x^B \rangle_{x\in\mathbb{N}}$ will be bounded by a very slow-growing order function, making use of the assumption $A \leq_{\mathrm{SJT}} B$. However, we do not control the computable bound on the size of $\langle V_y \rangle_{y\in\mathbb{N}}$; this bound can grow very quickly, as the assumption is only that B is jump-traceable, not strongly so. Observe that in this direction, A being jump-traceable follows from B being jump-traceable and $A \leq_{\mathrm{SJT}} B$, since the latter implies $A \leq_{\mathrm{JT}} B$.

We will need a combinatorial lemma.

LEMMA 2.3. *Suppose that $T \subset \omega^{<\omega}$ is a finite tree, and $v_0, \ldots, v_{n-1} \in T$ are pairwise distinct. If each v_i has at least two children in T, then T has at least $n + 1$ leaves.*

This can be proved by induction on n; we omit the details. The extreme case is when the nodes v_i form a chain (they are all comparable).

Fix a benign cost function $\mathbf{c}$. There is an ω-c.a. function f such that $c(f(n)) < 2^{-n}$ for all n. There is a computable binary function, also denoted f, such that $\lim_s f(n, s) = f(n)$ for all n, and such that for each n, $|\{f(n, s) : s \in \omega\}| \leq g(n)$, where g is some total computable function. We may assume that $f(n, s)$ is non-decreasing in both n and s.

Let us discuss the structure of the argument. It has two steps:

(1) Enumeration of the functionals Φ and Ψ.
(2) Using the results of (1) to define the approximation of A.

The first step is a computable construction, and as we later discuss, it has to be uniform in the indices for Φ and Ψ. The second step is computable in B, and uses the correct indices of Φ and Ψ, so is non-uniform in that sense too.

As discussed, the argument is adaptive, in that in the definition of Φ and Ψ, we will make use of a trace $\langle V_y \rangle_{y\in\mathbb{N}}$ that traces Ψ^B, and an oracle trace $\langle U_x^- \rangle_{x\in\mathbb{N}}$ such

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

that $\langle U_x^B \rangle_{x\in\mathbb{N}}$ traces Φ^A. Working with partial functions allows us to achieve this using the recursion theorem (as in [6]), however the method of a universal trace would also work here (see, e.g., [18]).

The general idea (as in [8]) is to *test* potential initial segments of A by setting $\Phi^\sigma(x) = \sigma$ for all σ of some particular length ℓ, on a variety of locations (inputs) x. When $B = \emptyset$, we can observe the result of such a test by seeing which σ are enumerated into U_x; these values are preferred to others. The test is more useful when the bound on $|U_x|$ is smaller, for example, when $|U_x| = 1$ we know that the unique σ that appears in U_x must be an initial segment of A. We will start by testing the length $\ell = f(n)$ on inputs x with $|U_x| \leq n$. Of course, we do not know the final value $f(n)$, so we will need to test all lengths $f(n, s)$, for all s, on such inputs x. Once we test a certain length on x, we cannot use the same input x again for other lengths, so we need to ensure that we have sufficiently many x with $|U_x| \leq n$. Luckily, we control the bound on the size of U_x, and we also know that we will encounter at most $g(n)$ many values $f(n, s)$, so this allows us to reserve sufficiently many x for this purpose.

The heart of the construction is a *promotion* process for various lengths that we test. When $|U_x| > 1$, we may get more than one result, σ_0 and σ_1, both enumerated in U_x, and we need to devise a procedure that will tell us which of these we will believe more. In such a situation, we will (very roughly) promote the length $\ell = |\sigma_0| = |\sigma_1|$ from level n testing to level $(n-1)$-testing, which gives us stronger results (the actual details will be a bit different). To know that we have sufficiently many locations x for these extra tests, we need to bound the number of lengths promoted from level n to level $n-1$. To do this, some of our tests will not use a single location x, but many such locations, and we will only trust some σ if it appears in *all* U_x on which it was tested.

This process of promotion is where the construction becomes adaptive: based on enumerations into various U_x, we decide to make further tests, defining more values of Φ^A. The overall structure of our construction, when $B \neq \emptyset$, is the same, except that, as discussed, we cannot examine U_x^B during the construction. Rather, for any oracle Y (uniformly), we will define Ψ^Y while examining $\langle U_x^Y \rangle_{x\in\mathbb{N}}$. As mentioned, the whole construction of Φ and Ψ will also, adaptively, examine values in another trace $\langle V_y \rangle_{y\in\mathbb{N}}$ (that has no oracle, and traces the "correct" Ψ^B).

For the definition of Ψ, we computably partition $\mathbb{N}$ into sets R_x (for $x \in \mathbb{N}$) and Q_n (for $n \in \mathbb{N}$), such that $|R_x| = x$ and $|Q_n| = n$. We will shortly see how using the recursion theorem, during the construction, we have a c.e. trace $\langle V_y \rangle_{y\in\mathbb{N}}$ for Ψ^B, and a computable bound h of $\langle V_y \rangle_{y\in\mathbb{N}}$.

Given h, for the purpose of defining Φ, we now computably partition $\mathbb{N}$ to sets I_n and J_n, so that:

$$|I_n| = g(n) + \sum \{h(y) : y \in Q_{n+1}\},$$

and $|J_n| = 2^{r_n}$, where

$$r_n = \sum \Big\{h(y) : y \in \bigcup_{x\in I_n} R_x\Big\}.$$

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

CLAIM 1.4.1 (Number o and trace $\langle U^-_x\rangle_{x\in\mathbb{N}}$). *During the construction, we have a number o and an oracle-c.e. trace $\langle U^-_x\rangle_{x\in\mathbb{N}}$ such that for all $n > o$, for all $x \in I_n \cup J_n$,*

- *for all Y, $|U^Y_x| \leq n$;*
- *if $\Phi^A(x)$ is defined, then $\Phi^A(x) \in U^B_x$.*

It would seem that $\langle V_y\rangle_{y\in\mathbb{N}}$, h and $\langle U^-_x\rangle_{x\in\mathbb{N}}$ are provided by the recursion theorem in a straightforward way. However, we note that h depends on the index of Ψ, and that in general, the assumption $A \leq_{\mathrm{SJT}} B$ does not allow us to uniformly, given an order function k, obtain a k-bounded trace of J^A (using a universal trace we can obtain such a trace but which may omit tracing finitely many values of Φ^A, which will amount to the same thing as we do here). We do the following.

We start with a c.e. trace of J^B, bounded by some order function $\tilde{h}$. For each e, let $h^e(y) = \tilde{h}(e, y)$. Assuming that $J^Y(e, y) = \Phi^Y_e(y)$ for all Y, e, and y, this gives us, uniformly in e, the index of a c.e. trace $\langle V^e_y\rangle_{y\in\mathbb{N}}$ of Ψ^B_e bounded by h^e.

The definition above then gives us, for each e, a partition of $\mathbb{N}$ to sets I^e_n and J^e_n ($n \in \mathbb{N}$), using the bound h^e. This allows us to compute an order function k such that for all d, e, $n > \langle d, e\rangle$ and $x \in I^e_n \cup J^e_n$, we have $k(d, x) \leq n$. Fix a k-bounded, B-c.e. trace $\left\langle \tilde{U}^B_x\right\rangle_{x\in\mathbb{N}}$ of J^A; we may assume that for any oracle Y, $\left\langle \tilde{U}^Y_x\right\rangle_{x\in\mathbb{N}}$ is k-bounded. This allows us to compute, uniformly in d and e, a c.e. index of an oracle trace $\langle U^-_x\rangle_{x\in\mathbb{N}}$ that satisfies the required conditions when $\Psi = \Psi_e$ and $\Phi = \Phi_d$, where $o = \langle d, e\rangle$.

By the recursion theorem, during the construction we know the indices d and e, and so the number o, the sets I_n and J_n, and c.e. indices of $\langle V_y\rangle_{y\in\mathbb{N}}$ and $\langle U^-_x\rangle_{x\in\mathbb{N}}$, as promised. This shows Claim 2.3.

We are ready to give the details of the construction of Φ and Ψ. The definition of Ψ on the sets R_x is immediate:

(i) For all $n > o$, for all $x \in I_n$, for every oracle Y, for all $i < n$, if ℓ is the ith number enumerated into U^Y_x, and y is the ith element of R_x, then we define $\Psi^Y(y) = \ell$.

Note that we may assume that $x \in I_n$ implies $x \geq n$, so setting $|R_x| = x$ is certainly sufficient to make these definitions, recalling that $|U^Y_x| \leq n$ when $n > 0$ and $x \in I_n$.

We will have a module for each $n > o$. This module can perform the following actions:

- test a length on I_n;
- test a string on J_n;
- for any oracle Y, declare that Y believes that a length ℓ should be promoted;
- promote some lengths to the $(n-1)$-module.

We describe what each of these means, when these actions are performed, and argue that we can indeed perform these actions.

2.3.1. Promoting lengths Let $n > o$.

(ii) The n-module promotes every $\ell \in \bigcup_{y\in Q_n} V_y$.

This definition applies even to the smallest module $n = o + 1$, however, for this module, no actual action will be taken.

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

2.3.2. Testing lengths on I_n

(iii) At stage 0, the n-module tests $f(n,0)$ on I_n. At stage $s>0$, if $f(n,s)\neq f(n,s-1)$, the n-module tests $f(n,s)$ on I_n. Finally, if at some stage s, the $(n+1)$-module promotes a length ℓ, then the n-modules tests ℓ on I_n.

(iv) Testing a length ℓ on I_n at stage s means: choose some unused $x\in I_n$, and define $\Psi^\sigma(x)=\sigma$ for all binary strings σ of length ℓ.

We need to argue that such a fresh x can always be chosen; that is, we need to show that the n-module will test at most $|I_n|$ many lengths during the entire construction. The number $g(n)$ bounds the number of lengths $f(n,s)$ ever tested on I_n. By (ii), at most $\sum\{h(y) : y\in Q_{n+1}\}$ many lengths are every promoted by the $(n+1)$-module. So the definition of $|I_n|$ ensures that it is large enough.

2.3.3. Testing strings on J_n

(v) The n-module tests on J_n every string in $\bigcup_{x\in I_n}\bigcup_{y\in R_x} V_y$.

Note that the n-module will test at most r_n many strings during the construction. Since $|J_n|=2^{r_n}$, for the purposes of such testing, we identify J_n with $\{0,1\}^{r_n}$, which we think of as a hypercube of side length 2 and dimension r_n. We let $d=0,1,\dots,r_n-1$ denote the axes of this hypercube; for $d<r_n$ we let $J_n(d)=\{\tau\in J_n : \tau(d)=0\}$. This is one half of a split of the hypercube into two pieces orthogonal to axis d.

(vi) Testing a string σ on J_n means the following. We choose an unused axis d and define $\Phi^\sigma(x)=\sigma$ for all unused $x\in J_n(d)$.

Now (v) ensures that indeed, there will always be an unused axis to choose.

2.3.4. Seeking promotion Let $n>o$ and let Y be an oracle.

(vii) We say that Y *confirms* a string σ tested on J_n if $\sigma\in U_x^Y$ for every $x\in J_n$ on which we defined $\Phi^\sigma(x)=\sigma$.

(viii) Let s be a stage; let ℓ be the longest length which Y believed should be promoted prior to stage s. Suppose that there are two strings, σ_0 and σ_1, of the same length, such that:
- $\sigma_0\upharpoonright\ell=\sigma_1\upharpoonright\ell$;
- Y confirms both σ_0 and σ_1 on J_n at stage s.

Then we say that Y believes that the length $|\sigma_0|=|\sigma_1|$ should be promoted by the n-module.

(ix) The action taken then is: if m is the ith length that Y believes should be promoted by the n-module, and $i<n$, and y is the ith element of Q_n, then we define $\Psi^Y(y)=m$.

We remark that it is quite possible that some oracles will believe that more than n many lengths should be promoted by the n-module. However this will not be the case for $Y=B$, as the next claim states.

We presented the construction in a way that makes it easier to show that it can indeed be performed. It would be good, though, to follow the steps in order:

(1) A length $\ell=f(n,s)$ is tested on I_n; this defines Φ on an element of I_n ((iii) and (iv) above).

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

(2) This prompts strings of length ℓ to appear in U_x^Y, and we make a definition of Ψ on an element of $y \in R_x$ (i).
(3) This prompts some strings to appear in V_y; in turn, these strings are tested on J_n ((v) and (vi)).
(4) This prompts strings to show up in various U_x^Y (for $x \in J_n$), and Y believes that some length ℓ should be promoted; a definition of $\Psi^Y(y)$ for some $y \in Q_n$ is made (with value ℓ, not a string of length ℓ) ((vii)–(ix)).
(5) This prompts lengths to show up in V_y, and the n-module promotes them (ii). We are now back to step (1) for the $(n-1)$-module for this promoted length.

This completes the description of the construction of Φ and Ψ.

Claim 1.4.2. *For each $n > o$, B believes at most $n-1$ lengths should be promoted by the n-module.*

Proof. Suppose not, and fix lengths $0 = \ell_0 < \ell_1 < \cdots < \ell_n$ such that for $i > 0$, B believes that ℓ_i is to be promoted by the n module. For each $i > 0$, fix strings σ_0^i and σ_1^i on the basis of which B believes that ℓ_i should be promoted by the n module.

By construction, B believes that ℓ_i should be promoted before it believes that ℓ_{i+1} should be promoted, and thus $\sigma_0^{i+1}\restriction_{\ell_i} = \sigma_1^{i+1}\restriction_{\ell_i}$ for $i > 0$. Clearly this also holds for $i = 0$. Now define the following sequence of sets (note that our subscripts are decreasing):

- $Z_n = \{\sigma_0^n, \sigma_1^n\}$;
- for $0 < i < n$,

$$Z_i = Z_{i+1} \cup (\{\sigma_0^i, \sigma_1^i\} \setminus \{\sigma\restriction_{\ell_i} : \sigma \in Z_{i+1}\}).$$

Note that each Z_i is an antichain, and $\{\sigma_0^i, \sigma_1^i\} \subseteq \{\sigma\restriction_{\ell_i} : \sigma \in Z_i\}$ by construction. Let $T = \{\sigma\restriction_{\ell_i} : \sigma \in Z_1 \ \&\ i \leq n\}$, which we think of as a tree. The leaves of T are precisely the elements of Z_1.

For $i < n$, let $v_i = \sigma_0^{i+1}\restriction_{\ell_i} = \sigma_1^{i+1}\restriction_{\ell_i}$. Then the v_i are pairwise distinct because they all have different lengths, and each has at least two children in T (namely, σ_0^{i+1} and σ_1^{i+1}). By Lemma 2.3, $|Z_1| \geq n+1$.

Let D be the set of axes chosen for various $\sigma \in Z_1$, and define $\tau \in J_n$ by

$$\tau(d) = \begin{cases} 0 & d \in D, \\ 1 & d \notin D. \end{cases}$$

Observe that $\tau \in J_n(d) \iff d \in D$.

Recall that our construction will define $\Phi^\sigma(\tau) = \sigma$ for all $\sigma \in Z_1$ unless $\Phi^\sigma(\tau)$ was already defined to be something else. But no $\sigma' \notin Z_1$ could have caused such a previous definition, as such σ' will have an axis $d \notin D$, and so will not seek to make a definition at τ. And $\sigma' \in Z_1 \setminus \{\sigma\}$ will not do this, as they will only seek to make a definition for $\Phi^{\sigma'}(\tau)$, and σ' and σ are incomparable as Z_1 is an antichain.

Thus $\Phi^\sigma(\tau) = \sigma$ for all $\sigma \in Z_1$. And by the definition of promotion and confirmation, we have $Z_1 \subseteq U_\tau^B$, contradicting $|U_\tau^B| \leq n$. ⊣

It follows that for every length ℓ which B believes should be promoted by the n-module, there is a corresponding $y \in Q_n$ such that $\Psi^B(y) = \ell$. Since $\langle V_y \rangle_{y \in \mathbb{N}}$ traces Ψ^B, $\ell \in V_y$. Thus, every length which B believes should be promoted is eventually promoted.

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

We now proceed to the second part of the argument, in which we fix $Y = B$; the following construction is computable relative to B.

Let $\hat{\ell}$ be the longest length which B believes the $o+1$ module should promote. Nonuniformly, fix $A\upharpoonright_{\hat{\ell}}$. Let $L(n,s)$ be the set of lengths being tested by the n module at stage s (tested on I_n). At a stage s, define a partial sequence σ_n^s for $o \leq n \leq s$ recursively:

- $\sigma_o^s = A\upharpoonright_{\hat{\ell}}$.
- Given σ_n^s, if there is some length which B believes should be promoted by the $n+1$ module at stage s, but which is not yet promoted, then leave σ_{n+1}^s undefined. Otherwise, define σ_{n+1}^s to be a string ρ extending σ_n^s with $|\rho| = \max L(n+1,s)$, and such that for each $\ell \in L(n+1,s)$, $\rho\upharpoonright_\ell$ is confirmed by B at $n+1$ by stage s, if such a string ρ exists.

Claim 1.4.3. *There is at most one possible choice for σ_n^s.*

Proof. For $n = o$, this is immediate.

For $n > o$, suppose there were two distinct strings ρ_0 and ρ_1 which are appropriate to pick for σ_n^s. Fix $\ell \in L(n,s)$ least with $\rho_0\upharpoonright_\ell \neq \rho_1\upharpoonright_\ell$. Then $\rho_0\upharpoonright_\ell, \rho_1\upharpoonright_\ell$ witness that B believes that ℓ should be promoted at stage s, and $\ell > |\sigma_{n-1}^s|$, as ρ_0 and ρ_1 both extend σ_{n-1}^s. This contradicts $|\sigma_{n-1}^s| = \max L(n-1,s)$ (or contradicts the definition of $\hat{\ell}$ if $n = o+1$). ⊣

Claim 1.4.4. *Let $\ell_n = \max \bigcup_s L(n,s)$ for $n > o+1$, and $\ell_{o+1} = \hat{\ell}$. Then $A\upharpoonright_{\ell_n} = \lim_s \sigma_n^s$ for $n > o$.*

Proof. Induction on n. The case $n = o+1$ is immediate.

For $n > o$, first observe that ℓ_{n-1} is either a length promoted by the n module (and so eventually an element of $L(n,s)$) or is $f(n-1,s) \leq f(n,s)$ for some s, and so is bounded by an element of $L(n,s)$. Thus $\ell_{n-1} \leq \ell_n$.

Now fix s_0 sufficiently large such that $\sigma_m^s = A\upharpoonright_{\ell_m}$ for all $m < n$ and $s \geq s_0$, and such that $L(n,s_0) = \bigcup_s L(n,s)$. As Φ^A is traced by $\langle U_x^B\rangle_{x\in\mathbb{N}}$, there is a stage $s_1 \geq s_0$ such that each $A\upharpoonright_\ell$ for $\ell \in L(n,s_0)$ is confirmed at n. Then $A\upharpoonright_{\ell_n}$ is a possible choice for σ_n^s for every $s \geq s_1$, and thus is σ_n^s. ⊣

Define a sequence of stages $(s_t)_{t\in\omega}$ as follows:

- $s_0 = o$.
- Given s_t, s_{t+1} is the least $s > s_t$ such that for every n with $o < n \leq t$, σ_n^s exists.

Define $A_t = \sigma_t^{s_t}$.

Claim 1.4.5. *The total* **c***-cost of the approximation $\langle A_t\rangle$ is finite.*

Proof. Suppose that $o < n \leq t$ and $A_t(z) \neq A_t(z+1)$ for some z with $c(z,t) \geq 2^{-n}$. As $c(z,s_t) \geq c(z,t)$, $z < f(n,s_t) \in L(n,s_t)$. Thus $\sigma_n^{s_t} \neq \sigma_n^{s_{t+1}}$. Fix m least with $\sigma_m^{s_t} \neq \sigma_m^{s_{t+1}}$. Fix $\ell \in L(m,s_t)$ least with $\sigma_m^{s_t}\upharpoonright_\ell \neq \sigma_m^{s_{t+1}}\upharpoonright_\ell$. If no length less than ℓ and greater than $\max L(m-1,s_t)$ is promoted by the m module at a stage $s \in (s_t, s_{t+1}]$, then these witness the promotion of ℓ at stage s_{t+1}, and $\ell > \max L(m-1,s_t)$, as both $\sigma_m^{s_t}$ and $\sigma_m^{s_{t+1}}$ extend $\sigma_{m-1}^{s_t} = \sigma_{m-1}^{s_{t+1}}$. So whenever there is such a z, n, and t, there is a promotion by an m module for $m \leq n$ at a stage after s_t.

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

There can be at most $\sum_{o<m\leq n} m-1 < n^2$ such promotions over the entire construction. Thus we can bound the total $\mathbf{c}$-cost of $\langle A_t \rangle$ by

$$\sum_n n^2 \cdot 2^{-n} < \infty.$$

⊣

This completes the proof of Theorem 1.4.

§3. Computing with random sets. In this section we give a proof of Theorem 1.5. First we discuss the notions of Demuth randomness and weak Demuth randomness; the latter appears in the statement of the theorem, while the former will play a role in its proof. We have the implications

$$\text{Demuth random} \Rightarrow \text{weakly Demuth random} \Rightarrow \text{ML-random}.$$

For further background, see [28, Section 3.6] on Demuth randomness, and in particular why this property is still compatible with being Δ^0_2. See [3] for weak Demuth randomness.

The general idea is that a Demuth test is a sequence (G_m) of effectively open (Σ^0_1) sets with $\boldsymbol{\lambda} G_m \leq 2^{-n}$, but unlike an ML-test, the sets G_m are not *uniformly* c.e. Rather, we think of the sets being given to us in stages, where from time to time, we are allowed to empty a component of the test and restart its enumeration. Each component is restarted only finitely many times, and in fact we require that there is a computable function bounding the number of times each component is restarted. For the formal definition, recall again that a function $f : \mathbb{N} \to \mathbb{N}$ is ω-c.a. if it has a computable approximation $f(m,s)$ such that the number of mind-changes $\#\{s : f(m,s) \neq f(m,s+1)\}$ is bounded by $g(m)$, where g is some computable function.

Definition 3.1. Let (W_e) be an admissible listing of the effectively open subsets of Cantor space. A *Demuth test* is a sequence $\langle G_m \rangle_{m\in\mathbb{N}}$ satisfying:

- for all m, $\boldsymbol{\lambda} G_m \leq 2^{-m}$;
- there is an ω-c.a. function p such that $G_m = W_{p(m)}$ for all m.

A set $Y \in 2^{\mathbb{N}}$ is *captured* by a Demuth test (G_m) if $Y \in G_m$ for infinitely many m. Otherwise, we say that Y *passes* the test. A set is *Demuth random* if it passes all Demuth tests.

If (G_m) is a Demuth test, witnessed by an ω-c.a. function p, and $p(m,s)$ is a computable approximation of p, then we write $G_m[s]$ for the clopen set $W_{p(m,s),s}$: at stage s, we guess that $G_m = W_{p(m,s)}$, and we let $G_m[s]$ be the result of enumerating that effectively open set for s many steps. We may assume that for all m and s, $\boldsymbol{\lambda} G_m[s] \leq 2^{-m}$: we can stop enumerating clopen subsets into the component if we see that its measure will exceed 2^{-m}. Note that in that case, we know that $p(m,s) \neq p(m)$, so we can wait for a new version of G_m to be started.

Thus, an "index-free" equivalent definition of a Demuth test (G_m) is: there is a computable array $(G_{m,s})$ of clopen sets such that for all m and s, $\boldsymbol{\lambda} G_{m,s} \leq 2^{-m}$, and there is a computable function g such that for all m, $\#\{s : G_{m,s} \not\subseteq G_{m,s+1}\}$ is bounded by $g(m)$; and $G_m = \bigcup_{s>t(m)} G_{m,s}$, where $t = t(m)$ is any stage sufficiently late so that for all $s \geq t$, $G_{m,s} \subseteq G_{m,s+1}$.

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

Note that the capturing condition is the Solovay one, rather than requiring $Y \in \bigcap_m G_m$. This is because in Definition 3.1, we do not require that the test be *nested*, meaning $G_{m+1} \subseteq G_m$ for all m. This is not an issue when considering ML-randomness; every non-nested ML test (U_m) can be transformed into a nested one by replacing U_m by $\bigcup_{n>m} U_n$. Applying this transformation to a Demuth test will usually not result in a Demuth test.

DEFINITION 3.2. A set Y is *weakly Demuth random* if it passes every nested Demuth test.

An equivalent definition is: for every Demuth test (G_m) (nested or not), $Y \notin \bigcap_m G_m$. That is, rather than restricting to nested tests, we can replace the passing condition. To see this, given a Demuth test (G_m), we let $H_m = \bigcap_{n \leq m} G_n$; then (H_m) is a nested Demuth test that captures all $Y \in \bigcap_m G_m$.

3.1. A proof of (a) $\Rightarrow$(b) of Theorem 1.5. In this direction, we only need the assumption that B is K-trivial. For each one of the five classes $\mathcal{C}$ listed in Theorem 1.5, let $(\mathrm{b})_{\mathcal{C}}$ indicate item (b) of the theorem as applied to this class: the statement that for all ML-random $Y \in \mathcal{C}$, $A \leq_{\mathrm{T}} Y \oplus B$. Of course, if $\mathcal{C} \subseteq \mathcal{D}$ then $(\mathrm{b})_{\mathcal{D}}$ implies $(\mathrm{b})_{\mathcal{C}}$. We note that among the five classes, the first, the class of sets that are not weakly Demuth random, is the largest. Namely, a weakly Demuth random set cannot be superhigh (this is a result of Kučera and Nies [23, Corollary 3.6]) and cannot be ω-c.a. (this is essentially by definition; if Y is ω-c.a. then $(\{Y\restriction_m\})$ is a nested Demuth test capturing Y). We note that every superlow set is ω-c.a., and that every set Ω_R for an infinite computable R is ω-c.a. as well.

Thus, in this direction, it suffices to show that (a) implies $(\mathrm{b})_{\mathcal{C}}$ where $\mathcal{C}$ is the class of sets that are not weakly Demuth random.

Hirschfeldt and Miller showed that for each null Π^0_2 class $\mathcal{H} \subseteq 2^\omega$, there is a cost function $\mathbf{c}$ such that for each A, $Y \in 2^\omega$, if $A \models \mathbf{c}$ and $Y \in \mathsf{MLR} \cap \mathcal{H}$ then $A \leq_T Y$ (see [28, Proof of 5.3.15] for a proof of this otherwise unpublished result). For the direction under discussion, we need to show that when $\mathcal{H}$ is the class of sets captured by some nested Demuth test, then the associated cost function $\mathbf{c}$ is benign, and that the Hirschfeldt–Miller result partially relativises as necessary. (See also the post on weak Demuth randomness by Kučera and Nies in [30].)

Let (G_m) be a nested Demuth test, with approximation $(G_{m,s})$ as discussed above. Note that we may assume that for each m and s, $G_{m+1,s} \subseteq G_{m,s}$. For $k < t$ we let:

- $r(k,t)$ be the smallest m such that for some s with $k < s \leq t$, $G_{m,s-1} \not\subseteq G_{m,s}$ (the smallest m such that the mth component of the test was restarted at some stage between k and t).
- $V_{k,t} = \bigcup_{k<s\leq t} G_{r(k,s),s}$.
- $\mathbf{c}(k,t) = \lambda V_{k,t}$.

By definition, $V_{k,t} \subseteq V_{k,t+1}$, so $\mathbf{c}$ is increasing in the second variable. Also, $V_{k,t} \supseteq V_{k+1,t}$: we have $r(k,s) \leq r(k+1,s)$, and so $G_{r(k,s),s} \supseteq G_{r(k+1,s),s}$. Hence, $\mathbf{c}$ is decreasing in the first variable, so $\mathbf{c}$ is monotonic.

Note that if no component $G_{m'}$ for $m' \leq m$ is restarted after stage k, then $r(k,s) > m$ for all $s > k$, implying that $G_{r(k,s),s} \subseteq G_{m,s}$ for all $s > k$, and so $V_{k,t} \subseteq G_m$ for all $t > k$.

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

Suppose that $\mathbf{c}(k,t) > 2^{-m}$. Since $\boldsymbol{\lambda} G_m \leq 2^{-m}$, this implies that $V_{k,t} \not\subseteq G_m$, so some component m' is restarted between stages k and t. Since there is a computable bound on the number of times each component is restarted, we see that the cost function $\mathbf{c}$ is benign.

We also observe that for all k, $\bigcap_m G_m \subseteq \bigcup_{t>k} V_{k,t}$. This is because $\lim_t r(k,t) = m$ where m is smallest such that the mth component is restarted after stage k, and so $G_m \subseteq \bigcup_{t>k} V_{k,t}$.

Suppose now that $A \models_B \mathbf{c}$; let (A_s) be a B-computable approximation with a finite total $\mathbf{c}$-cost. For each stage s, let x_s be the least such that $A_s(x) \neq A_{s+1}(x)$; we let

$$R_s = V_{x_s,s}.$$

Each set R_s is clopen, and the sequence (R_s) is computable in B. Since $\mathbf{c}(x_s, s) = \boldsymbol{\lambda} V_{x_s,s} = \boldsymbol{\lambda} R_s$, we get $\sum_s \boldsymbol{\lambda} R_s < \infty$. This means that (R_s) is a *Solovay test* relative to B.

Suppose that Y is ML-random and is captured by the nested Demuth test (G_m). Since B is assumed to be K-trivial, it is low for ML-random, and so Y is ML-random relative to B. Since (R_s) is a Solovay test relative to B, Y cannot be captured by (R_s), meaning that $Y \in R_s$ for only finitely many s. (Indeed, recall that a set is ML-random relative to B if and only if it passes all Solovay tests relative to B.)

Let s_0 be such that $Y \notin R_s$ for all $s \geq s_0$. We now show how to compute A given $Y \oplus B$. We are given some x and we find $A(x)$. To do that, using the oracle Y, find some $t \geq s_0$ such that $Y \in V_{k,t}$; we observed above that such t exists, since $Y \in \bigcap_m G_m$. We claim that $A(x) = A_t(x)$ (here the oracle B is used, since the approximation (A_t) is B-computable). Otherwise, $A_s(x) \neq A_{s+1}(x)$ for some $s \geq t$; so $R_s = V_{y,s}$ for some $y \leq x$. Since $V_{y,s} \supseteq V_{x,s} \supseteq V_{x,t}$, $Y \in R_s$, contrary to the assumption on s_0.

3.2. A proof of (b) $\Rightarrow$(a) of Theorem 1.5, cases (1)–(4). In this section we prove that (b)$_{\mathcal{C}}$ implies (a) for each of the first four classes $\mathcal{C}$ mentioned in Theorem 1.5: not weakly Demuth random, superhigh, ω-c.a., and superlow. The common property of these classes that is relevant here is that they are all Demuth-compatible, as defined in Nies [31].

DEFINITION 3.3. A class $\mathcal{C} \subseteq 2^{\mathbb{N}}$ is *Demuth-compatible* if for every Demuth test, there is some $Y \in \mathcal{C}$ that passes this test.

Using some methods from [15], Nies showed in [31, Section 4] that the class of superlow sets and the class of superhigh sets are Demuth-compatible. Every superlow set is ω-c.a., so the class of ω-c.a. sets is Demuth-compatible; every ω-c.a. is not weakly Demuth random, so the collection of sets that are not weakly Demuth random is also Demuth-compatible.

Thus, for the four classes under consideration, it suffices to show that if $\mathcal{C}$ is any Demuth-compatible class, then (b)$_{\mathcal{C}}$ implies (a). The argument combines methods from [3, 31].

We remark that for this implication, we can relax the requirement that A and B be K-trivial: it suffices to assume that A and B are each computable from some c.e., jump-traceable (equivalently, superlow by [27]) set. This is because every K-trivial set is jump-traceable, and every K-trivial set is computable from some

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

c.e. K-trivial set. Corollary 2.4 of [22] implies the following fact that we will use in the proof of the implication: if X is computable from a c.e., jump-traceable set, then X is *low for* BLR , as recalled next.

DEFINITION 3.4 (Cole and Simpson [34]). Let B be an oracle. A function f is *bounded limit-recursive* in B, denoted $\mathrm{BLR}(B)$, if it has a B-computable approximation $f(x,s)$, for which the number of mind-changes $\#\{s : f(x,s) \neq f(x,s+1)\}$ is bounded by a computable function.

We note that being bounded limit-recursive is a partial relativisation of the notion of ω-c.a.; the full relativisation would allow a B-computable bound on the number of mind-changes.

DEFINITION 3.5. We say that an oracle X is *low for* BLR if the $\mathrm{BLR}(X)$ functions are precisely the ω-c.a. functions.

Associated with the notion of partial relativisation of being ω-c.a. in Definition 3.4 is a partial relativisation of Demuth tests. A $\mathrm{Demuth}_{\mathrm{BLR}}\langle B\rangle$ test [3, Definition 1.7] is defined like a Demuth test relative to B, except that the function giving the $\Sigma^0_1(B)$ index of the mth component is $\mathrm{BLR}(B)$, rather than ω-c.a. relative to B. In terms of approximation, a $\mathrm{Demuth}_{\mathrm{BLR}}\langle B\rangle$ test is the limit of a B-computable approximation $(G_{m,s})$, with a *computable* bound on the number of times the mth component is restarted, rather than a B-computable one.

Fix an order function h, and sets A and B as described. We prove:

$(*)$ There is a Demuth test as follows. If $A \leq_{\mathrm{T}} Y \oplus B$ for some Y that passes the test, then J^A has an h-bounded, B-c.e. trace.

This suffices to show the desired implications.

We will show $(*)$ in a couple of steps. First, we will only consider a single Turing functional (reduction procedure) witnessing $A \leq_{\mathrm{T}} Y \oplus B$. We will show that there is a $\mathrm{Demuth}_{\mathrm{BLR}}\langle B\rangle$ test as required, and then cover it by an unrelativised Demuth test. Finally, we will use a universal functional to obtain $(*)$.

For the time being, fix a Turing functional Ψ.

We start by mostly following the proof of [31, Theorem 3.2]. For $m \in \mathbb{N}$ let $I_m = \left\{x \in \mathbb{N} : 2^m \leq h(x) < 2^{m+1}\right\}$, so each I_m is a finite interval and together they partition $\mathbb{N}$. For $\sigma \in 2^{<\mathbb{N}}$ let

$$U_\sigma = \left\{Z \in 2^{\mathbb{N}} : \sigma \preceq \Psi^{Z\oplus B}\right\};$$

the sets U_σ are $\Sigma^0_1(B)$, uniformly. For $m \in \mathbb{N}$ let

$$\sigma_m = \bigcup \left\{\sigma : (\exists x \in I_m)\ J^A(x)\downarrow \text{ with use } \sigma \prec A\right\}.$$

That is, σ_m is an initial segment $\sigma \prec A$ such that for all $x \in I_m$, $J^A(x)\downarrow \Leftrightarrow J^\sigma(x)\downarrow$. Note that since J^A is partial, the map $m \mapsto \sigma_m$ is not A-computable, rather, it is $\mathrm{BLR}(A)$: the number of mind-changes is bounded by $|I_m|$. By assumption on A, and the lowness fact mentioned above [22, Corollary 2.4], the map $m \mapsto \sigma_m$ is ω-c.a.; let $(\sigma_{m,s})$ be a computable approximation with a computable bound on the number of mind-changes. (For the current argument, we can allow the approximation $(\sigma_{m,s})$ to be B-computable rather than computable, but we need to keep the computable bound on the number of mind-changes.)

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

We recall some notation: for a $\Sigma^0_1(B)$ set V, and rational $\varepsilon > 0$, we let $V^{(\leq\varepsilon)}$ be the result of enumerating V (with oracle B), up to a point at which we see the measure exceeding ε. Thus, $V \subseteq V^{(\leq\varepsilon)}$, $\boldsymbol{\lambda}(V^{(\leq\varepsilon)}) \leq \varepsilon$, and $V = V^{(\leq\varepsilon)}$ if $\boldsymbol{\lambda} V \leq \varepsilon$. We then let

$$G_m = U_{\sigma_m}^{(\leq 2^{-m})},$$

and for each s we let $G_{m,s} = U_{\sigma_{m,s},s}^{(\leq 2^{-m})}$, the result of enumerating $U_{\sigma_{m,s},s}^{(\leq 2^{-m})}$ for s many steps. So the approximation $(G_{m,s})$ is B-computable, and the mth component is restarted only when $\sigma_{m,s} \neq \sigma_{m,s+1}$, for which we have a computable bound. Hence, (G_m) is a $\mathrm{Demuth}_{\mathrm{BLR}}\langle B\rangle$ test.

We claim that (G_m) is close to what is required in $(*)$: if $A = \Psi^{B\oplus Y}$ for some Y that passes the test (G_m), then J^A has an h-bounded B-c.e. trace. The argument is similar to that of [31, Theorem 3.2]. For each m, $x \in I_m$, and s, we enumerate $J^{\sigma_{m,s}}(x)$ into T_x if it is defined at stage s, and the measure of $G_{m,s}$ is exactly 2^{-m} (the stage s version of G_m is "full").

Suppose that y, z are distinct elements of T_x; suppose that we enumerate y into T_x at stage s (so $y = J^{\sigma_{m,s}}(x)$ and $G_{m,s}$ is full at stage s), and we enumerate z into T_x at some stage t. Since $y \neq z$, the strings $\sigma_{m,s}$ and $\sigma_{m,t}$ are distinct incomparable, which implies that $G_{m,s}$ and $G_{m,t}$ are disjoint. Since both $G_{m,s}$ and $G_{m,t}$ are full, $|T_x| \leq 2^m \leq h(x)$. Hence, the B-c.e. trace (T_x) is h-bounded. Suppose that $A = \Psi^{Y\oplus B}$ and that $Y \notin G_m$. This means that $\boldsymbol{\lambda}(G_m) = 2^{-m}$ (otherwise at some late point we would enumerate Y into G_m), indeed, for all but finitely many stages s, $\boldsymbol{\lambda} G_{m,s} = 2^{-m}$. Hence, for all $x \in I_m$, if $J^A(x)$ is defined then $J^A(x) \in T_x$. So if $A = \Psi^{B\oplus Y}$ for some Y that passes (G_m), then a finite modification of (T_x) traces J^A.

Toward $(*)$, we use the fact that since B is computable from a c.e., jump-traceable set, it is *low for* $\mathrm{Demuth}_{\mathrm{BLR}}$ [3, Theorem 1.8, Proposition 4.3]: for every $\mathrm{Demuth}_{\mathrm{BLR}}\langle B\rangle$ test (G_m), there is an unrelativised Demuth test (H_m) that *covers* (G_m) in the sense that every set passing (H_m) also passes (G_m). Hence, there is a Demuth test (H_m) with the same property as (G_m): if $A = \Psi^{B\oplus Y}$ for some Y that passes (H_m), then J^A has an h-bounded B-c.e. trace.

Finally, the argument of [31, Lemma 2.6] gives $(*)$. Let (Φ_e) be an effective listing of all Turing functionals, and let $\Psi^{0^e 1X} = \Phi_e^X$ for all e and X. Let (H_m) be a Demuth test obtained for this functional Ψ. The referenced lemma states that there is a Demuth test (S_m) such that for all Y, if Y passes (S_m) then for all e, $0^e 1\, Y$ passes (H_m). Then (S_m) is as required for $(*)$; this completes the proof of (b) $_{\mathcal{C}} \Rightarrow$(a) for all Demuth-compatible classes $\mathcal{C}$.

3.3. Proof of (b) $\Rightarrow$(a) of Theorem 1.5, case (5). We now consider the remaining case of Theorem 1.5: if $A \leq_{\mathrm{T}} B \oplus \Omega_R$ for every infinite computable set R, then $A \leq_{\mathrm{SJT}} B$. In fact, using this assumption we prove that $A \models_B \mathbf{c}$ for every benign cost function $\mathbf{c}$, and use Theorem 1.4, which applies since every K-trivial set is jump-traceable. We generally follow arguments in [17], which gives the result when $B = \emptyset$.

Let R be infinite and computable. For $s > n$ we let

$$k_s(n) = \lfloor -\log_2(\Omega_s - \Omega_n) \rfloor,$$

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

and

$$\mathbf{c}_{\Omega,R}(n,s) = 2^{-|R\cap k_s(n)|}$$

(see [17, Definition 6.3]). The cost function $\mathbf{c}_{\Omega,R}$ is benign [17, Proposition 9.2]. Further, if $\mathbf{c}$ is any benign cost function, then there is some infinite computable R such that for all n, $\lim_s \mathbf{c}(n,s) \leq^\times \lim_s \mathbf{c}_{\Omega,R}(n,s)$ [17, Proposition 9.3]. By [32, Theorem 3.4], this implies that if $A \models_B \mathbf{c}_{\Omega,R}$ then $A \models_B \mathbf{c}$. Hence, to prove the desired implication, the following suffices.

LEMMA 3.6. *Let A and B be K-trivial, and R be infinite and computable. If $A \leq_T B \oplus \Omega_R$ then $A \models_B \mathbf{c}_{\Omega,R}$.*

This is the required partial relativisation of [17, Lemma 6.9]. We will follow the proof of this lemma. As it is fairly long, we will not copy all the details; rather, we will give the overall structure of the proof, and indicate how it needs to be modified to accommodate $B \neq \emptyset$.

The proof has two main steps. We first prove Lemma 3.6 assuming that A is c.e. relative to B; we then show how to remove this extra assumption.

3.3.1. Proof of Lemma 3.6 assuming that A is c.e. in B We may assume that R is co-infinite; otherwise, $\mathbf{c}_{\Omega,R} =^\times \mathbf{c}_\Omega$, and since A is K-trivial, it obeys $\mathbf{c}_\Omega$ (with no need for B's help). For $n \in \mathbb{N}$ let

$$U_n = \bigcup_{s\geq n} \left[(\Omega_s)_{R^{\mathtt{C}}} \upharpoonright |R^{\mathtt{C}} \cap n|\right]$$

(the string in the definition is the result of erasing the bits in locations in R from the string $\Omega_s \upharpoonright n$); let $U_{n,s}$ be the stage s enumeration of U_n. Then $\boldsymbol{\lambda}(U_{n,s}) \leq^\times \mathbf{c}_{\Omega,R^{\mathtt{C}}}(n,s)$ and $\Omega_{R^{\mathtt{C}}} \in \bigcap_n U_n$. Following the notation in [17], let $Y = \Omega_R$ and $X = \Omega_{R^{\mathtt{C}}}$. Fix a B-computable functional Φ such that $A = \Phi^Y$. Fixing a B-computable enumeration (A_s) of A, we let E_s, the stage s error set, be the collection of Z such that Φ_s^Z lies strictly to the left of A_s, and $E = \bigcup_s E_s$ is the set of Z such that Φ^Z lies to the left of A. We let $Q_s = 2^\omega \times (2^\omega \setminus E_s)$ (and $Q = 2^\omega \times (2^\omega \setminus E)$). We perform the "ravenous sets" construction precisely as in [17], except that now the construction is computable in B. We obtain sets V_n^k, and let $V^k = \bigcup_n V_n^k$.

The main part of the verification that we need to discuss is that $(X, Y) \notin \bigcap_k V^k \cap Q$. The first step of this is the equivalence between that statement, and Q having positive density at (X, Y). When $B = \emptyset$ this is [5, Lemma 3.3] (quoted as [17, Fact 8.1]). The relativisation of this fact gives:

$(\otimes)_1$: Suppose that Y is random relative to B, P is a $\Pi^0_1(B)$-class, and $Y \in P$. Then P has density 0 at Y if and only if Y fails a B-difference test on P.

Here a B-difference test on P is a test of the form $(P \cap O_n)$, where (O_n) are uniformly $\Sigma^0_1(B)$ and $\boldsymbol{\lambda}(P \cap O_n) \leq 2^{-n}$. We also need the following relativisation of a characterisation of difference randomness from [13]: if Y is an ML-random set,

$(\otimes)_2$: Y passes all B-difference tests if and only if $Y \oplus B \geq_{\mathrm{T}} B'$.

Now we argue as follows. Since Y is ML-random and B is K-trivial, Y is B-random. Since B is K-trivial, it is low, i.e., $B' \equiv_{\mathrm{T}} \emptyset'$. Now Y is Turing incomplete (since R is co-infinite), that is, $Y \not\geq_{\mathrm{T}} \emptyset'$. By a result of Day and Miller [7], since B is K-trivial, $Y \oplus B \not\geq_{\mathrm{T}} \emptyset'$ (this result is not needed in [17]). By $(\otimes)_2$, Y passes all

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

B-difference tests. Since $Y \notin E$, by $(\otimes)_1$, $2^\omega \setminus E$ has positive density at Y (again we use the fact that Y is B-random, as B is K-trivial). It follows that Q has positive density at (X, Y). Again since (X, Y) is ML-random, it is B-random. By $(\otimes)_1$ again, $(X, Y) \notin \bigcap_k V^k \cap Q$, as required.

The rest of the verification in [17] goes through when relativising to B, finishing the proof of Lemma 3.6 when A is c.e. relative to B.

3.3.2. Proof of Lemma 3.6 without extra assumptions As in [17], this case follows from the special case above using two facts:

$(\otimes)_3$: If $A \leq_{\mathrm{T}} B \oplus \Omega_R$ then there is some B-c.e., K-trivial set D such that $A \leq_{\mathrm{T}} B \oplus D$ and $D \leq_{\mathrm{T}} B \oplus \Omega_R$.

$(\otimes)_4$: If D is K-trivial, $D \models_B \mathbf{c}_{\Omega,R}$ and $A \leq_{\mathrm{T}} D \oplus B$ then $A \models_B \mathbf{c}_{\Omega,R}$.

We start with the latter, following the proof in [17, Lemma 8.2] (which is similar to [16, Proposition 2.3]). Let ψ be the use of a computation of A from $D \oplus B$. The K-trivial degrees are closed under taking joins, so $D \oplus B$ is K-trivial. Hence [1, Lemma 2.5] applies, so $\Omega - \Omega_n \leq^{\times} \Omega - \Omega_{\psi(n)}$. The rest of the proof of [17, Lemma 8.2] follows without changes.

We turn to verify $(\otimes)_3$, which follows from a partial relativisation of [17, Theorem 3.1]. For this, we need a partial relativisation of [17, Lemma 3.2]:

$(\otimes)_5$: Suppose that A and B are K-trivial. There is a B-computable approximation (A_s) of A such that if X is ML-random and $A = \Phi(X, B)$ then for almost all n, if $A \upharpoonright n \preceq \Phi_s(X, B)$ then $A \upharpoonright n = A_t \upharpoonright n$ for all $t \geq s$.

This follows from the proof in [17], using a partially relativised version of the "main lemma" derived from the golden run construction [28, 5.5.1]; we need the prefix-free machine M to be B-computable, and the sequence of stages $(q(i))$ to be B-computable. Again, we follow the golden run construction without changes, relativising all to B. We use the fact that B is low for K, so A is K-trivial relative to B: $K^B(A \upharpoonright n) =^+ K(A \upharpoonright n) =^+ K(n) =^+ K^B(n)$. The added constants are incorporated into the construction.

This completes the proof of Lemma 3.6, and so of Theorem 1.5.

§4. Concluding remarks and open questions. In this section we mention some corollaries of our main theorems, discuss related results, and state some open questions.

The very first characterisation of strong jump-traceability was in terms of Kolmogorov complexity [12]. This extends to a characterisation of the weak reducibility $\leq_{\mathrm{SJT}}$. Informally, $A \leq_{\mathrm{SJT}} B$ if and only if the plain descriptive string complexity C^A is "almost" lower-bounded by C^B. The formal version follows.

PROPOSITION 4.1. *The following are equivalent for all sets A, B:*

(a) $A \leq_{SJT} B$;
(b) $\forall x\, [C^B(x) \leq^+ C^A(x) + h(C^A(x))]$ *for each order function h.*

Theorem 17 of [12] states this for the case that $B = \emptyset$ (also see [28, Corollary 8.4.32]). The proof gives the required partial relativisation, where we still almost lower-bound C^A (rather than $C^{A \oplus B}$), and the functions h are computable.

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

4.1. The structure of the SJT degrees. Not much is known about the degree structure given by $\leq_{\mathrm{SJT}}$. We can state the following.

PROPOSITION 4.2. *If B is jump traceable, then the set $\{A : A \leq_{SJT} B\}$ is countable.*

In particular, the $\leq_{\mathrm{SJT}}$-degree of B is countable, and it bounds only countably many $\leq_{\mathrm{SJT}}$-degrees.

PROOF. By Theorem 1.4, if B is jump traceable and $A \leq_{\mathrm{SJT}} B$, then $A \models_B \mathbf{c}$ for any benign cost function. In particular, $A \in \Delta^0_2(B)$, which is a countable set. ⊣

In contrast, not all initial segments of the degree structure induced by $\leq_{\mathrm{SJT}}$ are countable: Ng [24, Theorem 6.2.1] proved that the set $\{A : A \leq_{\mathrm{SJT}} \emptyset'\}$ contains a perfect class.

Like the K-trivials, the strongly jump-traceable degrees form an ideal. In particular, they are closed under taking joins. We ask the following question.

QUESTION 4.3. *Suppose that B is jump-traceable. Is the class $\{A : A \leq_{SJT} B\}$ closed under taking joins*?

By the argument of the proof of Theorem 1.6 (see Section 2.1), for an affirmative answer to Question 4.3, it would suffice to show that if $A_0, A_1 \leq_{\mathrm{SJT}} B$ and B is jump-traceable then $A_0 \oplus A_1 \leq_{\mathrm{JT}} B$. We could even hope for the following: there is a benign cost function $\mathbf{c}$ such that if B is jump-traceable and $A \models_B \mathbf{c}$ then A is jump-traceable (equivalently $A \leq_{\mathrm{JT}} B$). We remark that Question 4.3 for $B = \emptyset$ was first proved directly, without using cost functions (see [6] for the c.e. case); it would be interesting to see if ideas from that argument would be useful.

It is open whether the c.e. SJT-degrees are dense. We conjecture that for each c.e. set E that is not strongly jump traceable, there are c.e. sets $A, B \leq_{\mathrm{T}} E$ that are $\leq_{\mathrm{SJT}}$-incomparable. A certain obstacle to showing such structural results is the theorem of Ng [24] that the least degree in the c.e. SJT-degrees, given by the strongly-jump traceable c.e. sets, has a Π^0_4 complete index set. Examining the definition, one sees that the arithmetical complexity of the reducibility itself on the c.e. sets is also Π^0_4. In contrast, Turing reducibility is Σ^0_4, and the least degree is merely Σ^0_3.

4.2. Relationship with LR-reducibility. Recall the weak reducibility $\leq_{\mathrm{LR}}$ mentioned in the introduction: $A \leq_{\mathrm{LR}} B$ if $\mathsf{MLR}^B \subseteq \mathsf{MLR}^A$. Every strongly jump-traceable set is K-trivial, that is, if $A \leq_{\mathrm{SJT}} \emptyset$ then $A \leq_{\mathrm{LR}} \emptyset$.

QUESTION 4.4. *Does $A \leq_{SJT} B$ imply $A \leq_{LR} B$*?

Here Theorem 1.6 may be relevant: the relation $A \oplus B \leq_{\mathrm{LR}} B$ is much better understood than $A \leq_{\mathrm{LR}} B$, as it has a characterisation in terms of K-triviality. In particular, this characterisation implies that every $\leq_{\mathrm{LR}}$-degree is countable, even though some $\leq_{\mathrm{LR}}$-degrees bound uncountably many such degrees. In contrast, we do not know whether every $\leq_{\mathrm{SJT}}$-degree is countable.

In general, it would be interesting to give characterisations of other weak reducibilities using the three paradigms discussed in the introduction. In particular, we suggest the reducibilities $A \leq_{\mathrm{LR}} B$, $A \leq_{\mathrm{JT}} B$, and $A' \leq_{\mathrm{tt}} B'$ (see [28, Section 8.4]).

4.3. Relationship with ML-reducibility. The following relation is a main topic of the article [17].

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

DEFINITION 4.5. For sets $A, B \subseteq \mathbb{N}$, one writes $A \leq_{\mathrm{ML}} B$ if for every ML-random Y, $Y \geq_{\mathrm{T}} B$ implies $Y \geq_{\mathrm{T}} A$.

This relation is particularly useful in understanding the structure of the K-trivial Turing degrees. It satisfies the conditions for being a weak reducibility, except that it is not known to be an arithmetical relation, even when restricted to the K-trivials.

There are certain differences between the reducibilities $\leq_{\mathrm{ML}}$ and $\leq_{\mathrm{SJT}}$ on the K-trivial sets. For example, the least $\leq_{\mathrm{ML}}$-degree consists only of the computable sets, not all strongly jump-traceable sets (this shows that $\leq_{\mathrm{SJT}}$ does not imply $\leq_{\mathrm{ML}}$ on the K-trivials). Further, there is a greatest $\leq_{\mathrm{ML}}$-degree of K-trivials (called the "smart" K-trivials in [4, 17]). By the following proposition there is no greatest $\leq_{\mathrm{SJT}}$-degree among the K-trivials; in particular, $\leq_{\mathrm{ML}}$ does not imply $\leq_{\mathrm{SJT}}$ on the K-trivials.

PROPOSITION 4.6. *For each K-trivial set B, there is a c.e. K-trivial set $C \geq_T B$ such that $C \not\leq_{SJT} B$.*

PROOF. By [6], for some fixed computable function h there is a functional Ψ and K-trivial set A such that Ψ^A has no c.e. trace bounded by h. (Also see [28, 8.5.1] where this is shown for $h(n) = 0.5 \log\log n$.) Since there is a computable function p such that $\Psi^X(n) = J^X(p(n))$ for each X, n, we get a computable function g so that the statement holds for J and g instead of Ψ and h. Relativizing to B we can retain the same g, so for each B there is a set A that is K-trivial in B, such that $J^{\hat{C}}$ does not have a B-c.e. trace bounded by g, where $\hat{C} = A \oplus B$. In particular, $\hat{C} \not\leq_{\mathrm{SJT}} B$.

If B is K-trivial, it is low for K, and hence A is also K-trivial. Therefore $\hat{C}$ is K-trivial. To conclude the proof, take a K-trivial c.e. set $C \geq_T \hat{C}$. (For standard facts on K-trivial sets see, for instance, [28, Chapter 5].) ⊣

We conjecture that for each benign cost function $\mathbf{c}$, for each $B \models \mathbf{c}$, there is a c.e. set $A \models \mathbf{c}$ such that $A \not\leq_{\mathrm{SJT}} B$. This would strengthen Proposition 4.6.

By [17, Theorem 3.1], every ML-degree of a K-trivial set contains a c.e. set. By [8], every strongly jump-traceable set is computable from a c.e. strongly jump-traceable set. It is natural to ask the following question.

QUESTION 4.7. *Does every K-trivial $\leq_{SJT}$-degree contain a c.e. set?*

A reducibility strictly weaker than $\leq_{\mathrm{ML}}$ is $A \leq_{\omega\text{–ML}} B$: every ω-c.a. ML-random Turing above B is also above A. This reducibility is somewhat closer to $\leq_{\mathrm{SJT}}$. Theorem 1.5 for the class $\mathcal{C}$ being the ω-c.a. sets implies the following proposition.

PROPOSITION 4.8. *Let A and B be K-trivial.*

(1) *If $A \leq_{SJT} B$ then $A \leq_{\omega\text{–}ML} B$.*
(2) *$A \leq_{SJT} \emptyset$ if and only if $A \leq_{\omega\text{–}ML} \emptyset$.*

Finally, we mention a related notion. For $\mathcal{C} \subseteq \mathcal{P}(\mathbb{N})$ one defines $\mathcal{C}^{\diamond}$ as the class of c.e. sets that are Turing below each ML-random set in $\mathcal{C}$ [28, Section 8.5]. Theorem 1.5 implies the following corollary.

COROLLARY 4.9. *Let $\mathcal{C}$ be a nonempty class of ML-randoms that contains no weakly Demuth random. Then $\mathcal{C}^{\diamond}$ is downward closed under $\leq_{SJT}$.*

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

For instance, let $\mathcal{C} = \{\Omega_R\}$ for a co-infinite computable set R. This shows that the subideals, in the Turing sense, of the K-trivials considered in [16, 17] are actually closed downward under the weaker SJT-reducibility.

Acknowledgments. Some of this work was carried out at the Erwin Schrödinger Institute (ESI), Vienna, as part of the program on reverse mathematics in 2025.

Funding. All three authors gratefully acknowledge the support of the Marsden Fund of New Zealand.

REFERENCES

[1] G. Barmpalias and R. Downey, *Exact pairs for the ideal of the K-trivial sequences in the Turing degrees*. ***The Journal of Symbolic Logic***, vol. 79 (2014), no. 3, pp. 676–692.

[2] G. Barmpalias, J. S. Miller, and A. Nies, *Randomness notions and partial relativization*. ***Israel Journal of Mathematics***, vol. 191 (2012), no. 2, pp. 791–816.

[3] L. Bienvenu, R. Downey, N. Greenberg, A. Nies, and D. Turetsky, *Characterizing lowness for Demuth randomness*. ***The Journal of Symbolic Logic***, vol. 79 (2014), no. 2, pp. 526–569.

[4] L. Bienvenu, N. Greenberg, A. Kučera, A. Nies, and D. Turetsky, *Coherent randomness tests and computing the K-trivial sets*. ***Journal of the European Mathematical Society***, vol. 18 (2016), no. 4, pp. 773–812.

[5] L. Bienvenu, R. Hölzl, J. Miller, and A. Nies, *Denjoy, Demuth, and density*. ***Journal of Mathematical Logic***, vol. 14 (2014), no. 1, p. 1450004.

[6] P. Cholak, R. Downey, and N. Greenberg, *Strongly jump-traceability I: The computably enumerable case*. ***Advances in Mathematics***, vol. 217 (2008), pp. 2045–2074.

[7] A. R. Day and J. Miller, *Cupping with random sets*. ***Proceedings of the American Mathematical Society***, vol. 142 (2014), no. 8, pp. 2871–2879.

[8] D. Diamondstone, N. Greenberg, and D. Turetsky, *Inherent enumerability of strong jump-traceability*. ***Transactions of the American Mathematical Society***, vol. 367 (2015), no. 3, pp. 1771–1796.

[9] R. Downey and N. Greenberg, *Strong jump-traceability II: K-triviality*. ***Israel Journal of Mathematics***, vol. 191 (2012), no. 2, pp. 647–665.

[10] ———, *Pseudo-jump inversion, upper cone avoidance, and strong jump-traceability*. ***Advances in Mathematics***,

[11] R. Downey, D. Hirschfeldt, A. Nies, and F. Stephan, *Trivial reals*, ***Electronic Notes in Theoretical Computer Science***, vol. 66 (2002), no. 1, pp. 36–52. Elsevier.

[12] S. Figueira, A. Nies, and F. Stephan, *Lowness properties and approximations of the jump*. ***Annals of Pure and Applied Logic***, vol. 152 (2008), pp. 51–66.

[13] J. N. Y. Franklin and K. M. Ng, *Difference randomness*. ***Proceedings of the American Mathematical Society***, vol. 139 (2011), no. 1, pp. 345–360.

[14] N. Greenberg, *A random set which only computes strongly jump-traceable c.e. sets*. ***The Journal of Symbolic Logic***, vol. 76 (2011), no. 2, pp. 700–718.

[15] N. Greenberg, D. Hirschfeldt, and A. Nies, *Characterizing the strongly jump-traceable sets via randomness*. ***Advances in Mathematics***, vol. 231 (2012), nos. 3–4, pp. 2252–2293.

[16] N. Greenberg, J. S. Miller, and A. Nies, *Computing from projections of random points*. ***Journal of Mathematical Logic***, vol. 20 (2020), no. 1, p. 1950014.

[17] N. Greenberg, J. S. Miller, A. Nies, and D. Turetsky, *Martin–Löf reducibility and cost functions.* ***Israel Journal of Mathematics***, vol. 260 (2024), no. 1, pp. 261–301.

[18] N. Greenberg and A. Nies, *Benign cost functions and lowness properties*. ***The Journal of Symbolic Logic***, vol. 76 (2011), pp. 289–312.

[19] N. Greenberg and D. Turetsky, *Strong jump-traceability and Demuth randomness*. ***Proceedings of the London Mathematical Society***, vol. 108 (2014), pp. 738–779.

[20] ———, *Strong jump-traceability*. ***Bulletin of Symbolic Logic***, vol. 24 (2018), no. 2, pp. 147–164.

[21] B. Kjos-Hanssen, J. Miller, and R. Solomon, *Lowness notions, measure and domination*. ***Journal of the London Mathematical Society***, vol. 85 (2012), no. 3, pp. 869–888.

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press

[22] B. Kjos-Hanssen and A. Nies, *Superhighness*. ***Notre Dame Journal of Formal Logic***, vol. 50 (2009), no. 4, pp. 445–452.

[23] A. Kučera and A. Nies, *Demuth randomness and computational complexity*. ***Annals of Pure and Applied Logic***, vol. 162 (2011), pp. 504–513.

[24] K. M. Ng, ***Computability, traceability and beyond***. Doctoral dissertation, Victoria University of Wellington, 2009.

[25] ———, *Beyond strong jump traceability*. ***Proceedings of the London Mathematical Society***, vol. 102 (2010), no. 3, pp. 423–467.

[26] A. Nies, *Lowness properties and randomness*. ***Advances in Mathematics***, vol. 197 (2005), pp. 274–305.

[27] ———, *Reals which compute little*, ***Logic Colloquium '02***, (Z. Chatzidakis, P. Koepke, and W. Pohlers, editors), Lecture Notes Series, vol. 27, Association for Symbolic Logic, La Jolla, 2006, pp. 261–275.

[28] ———, ***Computability and Randomness***, Oxford Logic Guides, vol. 51, Oxford University Press, Oxford, 2009, 444 pp. Paperback version 2011.

[29] A. Nies, *Interactions of computability and randomness*, ***Proceedings of the International Congress of Mathematicians***, World Scientific, Singapore, 2010, pp. 30–57.

[30] A. Nies (editor), *Logic Blog 2011*, preprint, 2011. http://arxiv.org/abs/1403.5721.

[31] A. Nies, *Computably enumerable sets below random sets*. ***Annals of Pure and Applied Logic***, vol. 163 (2012), no. 11, pp. 1596–1610.

[32] A. Nies, *Calculus of cost functions*, ***The Incomputable*** (S. B. Cooper, and M. I. Soskova, editors), Springer, Berlin, Heidelberg, New York, 2017, pp. 183–216.

[33] P. Odifreddi, *Strong reducibilities*. ***Bulletin of the American Mathematical Society***, vol. 4 (1981), no. 1, pp. 37–86.

[34] S. Simpson and J. Cole, *Mass problems and hyperarithmeticity*. ***Journal of Mathematical Logic***, vol. 7 (2007), no. 2, pp. 125–143.

[35] S. Terwijn, ***Computability and measure***, Ph.D. thesis, The Institute for Logic, Language and Computation (ILLC), University of Amsterdam, 1998.

VICTORIA UNIVERSITY OF WELLINGTON
NEW ZEALAND
E-mail: greenberg@sms.vuw.ac.nz
E-mail: dan.turetsky@vuw.ac.nz

UNIVERSITY OF AUCKLAND
NEW ZEALAND
E-mail: andre@cs.auckland.ac.nz

https://doi.org/10.1017/jsl.2026.10228 Published online by Cambridge University Press