LEARNING FROM MULTIPLE SOURCES OF INACCURATE DATA

GANESH BALIGA*, SANJAY JAIN †, AND ARUN SHARMA‡

Abstract. Most theoretical models of inductive inference make the idealized assumption that the data available to a learner is from a *single* and *accurate* source. The subject of inaccuracies in data emanating from a single source has been addressed by several authors. The present paper argues in favor of a more realistic learning model in which data emanates from *multiple* sources, some or all of which may be *inaccurate*. Three kinds of inaccuracies are considered: spurious data (modeled as *noisy* texts), missing data (modeled as *incomplete* texts), and a mixture of spurious and missing data (modeled as *imperfect* texts).

Motivated by the above argument, the present paper introduces and theoretically analyzes a number of inference criteria in which a learning machine is fed data from multiple sources, some of which may be infected with inaccuracies. The learning situation modeled is the identification in the limit of programs from graphs of computable functions. The main parameters of the investigation are: kind of inaccuracy, total number of data sources, number of faulty data sources which produce data within an acceptable bound, and the bound on the number of errors allowed in the final hypothesis learned by the machine.

Sufficient conditions are determined under which, for the same kind of inaccuracy, for the same bound on the number of errors in the final hypothesis, and for the same bound on the number of inaccuracies, learning from multiple texts, some of which may be inaccurate, is equivalent to learning from a single inaccurate text.

The general problem of determining when learning from multiple inaccurate texts is a restriction over learning from a single inaccurate text turns out to be combinatorially very complex. Significant partial results are provided for this problem. Several results are also provided about conditions under which the detrimental effects of multiple texts can be overcome by either allowing more errors in the final hypothesis or by reducing the number of inaccuracies in the texts.

It is also shown that the usual hierarchies resulting from allowing extra errors in the final program (results in increased learning power) and allowing extra inaccuracies in the texts (results in decreased learning power) hold.

Finally, it is demonstrated that in the context of learning from multiple inaccurate texts, spurious data is better than missing data which in turn is better than a mixture of spurious and missing data.

Key words. inductive inference, machine learning, inaccurate data, multiple sources.

AMS subject classifications. 68T05, 68T, 68, 68Qxx, 68Q

1. Introduction.

A scenario in which an *algorithmic* learner attempts to learn its environment may be described thus. At any given time, some finite data about the environment is made available to the learner. The learner reacts to this finite information by conjecturing a hypothesis to explain the behavior of its environment. Availability of additional data may cause the learner to revise its old hypotheses. The learner is said to be successful just in case the sequence of hypotheses it conjectures stabilizes to a final hypothesis which correctly represents its environment.

The above model, generally referred to as *identification in the limit*, originated in the pioneering work of Putnam [19], Gold [12], and Solomonoff [25, 26]. More recently, this model has been the subject of numerous studies in computational learning theory (see, e.g., [13, 20, 10, 27, 1]). A problem with this model is the idealized assumption that the data available to the learner is from a *single* and *accurate* source. The subject of inaccuracies in the data available to a learning machine has been previously studied

* Computer Science Department, Rowan College of New Jersey, Mullica Hill, NJ 08024, USA. Email: baliga@gboro.rowan.edu

† Department of Information Systems and Computer Science, National University of Singapore, Singapore 0511, Republic of Singapore. Email: sanjay@iscs.nus.sg

‡ School of Computer Science and Engineering, The University of New South Wales, Sydney, NSW 2052, Australia. Email: arun@cse.unsw.edu.au

by Schäfer-Richter [23], Fulk and Jain [11], Osherson, Stob and Weinstein [17], Jain [14, 15]. Each of these studies, however, also makes the assumption that the data available to the learner is from a single source. The present paper argues that in realistic learning situations, data available to a learner is from *multiple* sources, some of which may be *inaccurate*. We discuss these issues in the context of a specific learning scenario, namely, scientific inquiry modeled as identification of programs from graphs of computable functions. Although we present our results in the context of this particular learning task, we note that some of our arguments and techniques can be applied to other learning situations, too.

Consider a scientist S investigating a real world phenomenon F. S performs experiments on F, noting the result of each experiment, while simultaneously conjecturing a succession of candidate explanations for F. A criterion of success is for S to eventually conjecture an explanation which S never gives up and which correctly explains F. Since we never measure a continuum of possibilities, we could treat S as performing discrete experiments x on F and receiving back experimental results $f(x)$. By using a suitable Gödel numbering we may treat f associated with F as a function from N, the set of natural numbers, into N. Also, assuming a suitable neo-mechanistic viewpoint about the universe, f is computable. A complete and predictive explanation of F, then, is just a computer program for computing f. Thus, algorithmic identification in the limit of programs for computable functions from their graph yields a plausible model for scientific inquiry.

Let us consider some common practices in scientific inquiry. Data is usually collected using different instruments, possibly at different places (for example, astronomers use data from different telescopes situated at different locations). In many cases, experimental errors may creep in or the instruments may simply be faulty. In some extreme cases, the same instrument may record conflicting readings at different times. Also, occasionally it may be infeasible to perform experiments (for example, determining the taste of cyanide). Moreover, experimental findings of one scientist are generally available to others. All this tends to suggest that often a scientist receives data from multiple sources, many of which are likely to be inaccurate. The present paper incorporates these observations in the standard learning model. We now proceed formally.

Section 2 presents the notation; Section 3 presents the preliminary notions about identification in the limit and inaccurate data. Section 4 introduces the main subject of this paper, viz., learning in the presence of multiple sources of inaccurate data. In this section, we also discuss some of our results informally. Section 5 presents our results with proofs.

2. Notation.

Recursion-theoretic concepts not explained below are treated in [22]. N denotes the set of natural numbers, $\{0, 1, 2, 3, \ldots\}$, and N^+ denotes the set of positive integers, $\{1, 2, 3, \ldots\}$. $\in$, $\subseteq$, and $\subset$ denote, respectively, membership, containment, and proper containment for sets.

We let e, i, j, k, l, m, n, r, s, t, u, v, w, x, y, and z, with or without decorations[1], range over N. We let a, b, c, with or without decorations, range over $N \cup \{*\}$.

$[m, n]$ denotes the set $\{x \in N \mid m \leq x \leq n\}$. We let S, with or without decorations, range over subsets of N and we let A, B, C, and D, with or without decorations, range over finite subsets of N. $\min(S)$ and $\max(S)$ respectively denote the minimum and maximum element in S ($\max(S)$ is undefined if S contains infinitely

[1] Decorations are subscripts, superscripts, primes and the like.

many elements). We take $\min(\emptyset)$ to be ∞ and $\max(\emptyset)$ to be 0. $\mathrm{card}(S)$ denotes the cardinality of S. $*$ denotes *unbounded* but *finite*. We let $(\forall n \in N)[n < * < \infty]$. So then, '$\mathrm{card}(S) \leq *$' means that $\mathrm{card}(S)$ is finite.

Let $\lambda x, y \boldsymbol{.} \langle x, y\rangle$ denote a fixed pairing function (a recursive, bijective mapping: $N \times N \rightarrow N$) [22]. $\lambda x, y \boldsymbol{.} \langle x, y\rangle$ and its inverses are useful to simulate the effect of having multiple argument functions. π_1 and π_2 are corresponding projection functions, i.e., $(\forall x, y)[\pi_1(\langle x, y\rangle) = x \ \wedge \ \pi_2(\langle x, y\rangle) = y]$.

η and ξ range over partial functions. For $a \in (N \cup \{*\})$, we say that η_1 is an a-variant of η_2 (written $\eta_1 =^a \eta_2$) iff $\mathrm{card}(\{x \mid \eta_1(x) \neq \eta_2(x)\}) \leq a$. Otherwise we say that η_1 is not an a-variant of η_2 (written $\eta_1 \neq^a \eta_2$). $\mathrm{domain}(\eta)$ and $\mathrm{range}(\eta)$ respectively denote the domain and range of partial function η. Let $A \subseteq N$, $c \in N$. We say that $\eta(A) = c$ iff, for all $x \in A$, $\eta(x) = c$. If S_1, S_2 are two sets, then $S_1 \boldsymbol{\Delta} S_2$ denotes $(S_1 - S_2) \cup (S_2 - S_1)$.

$\mathcal{R}$ denotes the class of all *recursive* functions of one variable, i.e., total computable functions with arguments and values from N. f, g, and h, with or without decorations, range over $\mathcal{R}$. $\mathcal{C}$ and $\mathcal{S}$, with or without decorations, range over subsets of $\mathcal{R}$.

We fix φ to be an *acceptable programming system* [21, 22, 16] for the partial recursive functions: $N \rightarrow N$. φ_i denotes the partial recursive function computed by φ-program with index i. W_i denotes $\mathrm{domain}(\varphi_i)$. W_i is, then, the r.e. set/language ($\subseteq N$) accepted (or equivalently, generated) by the φ-program i. We let Φ be an arbitrary Blum complexity measure [4] associated with acceptable programming system φ; such measures exist for any acceptable programming system [4]. $W_{i,s}$ denotes the set $\{x \mid x < s \ \wedge \ \Phi_i(x) \leq s\}$.

In some contexts p and q, with or without decorations range over programs. In other contexts p, q range over total recursive functions, with the range of p, q being interpreted as (indexes for) programs. In some contexts P, with or without decorations, ranges over programs. In other contexts, P ranges over sets of programs.

For any predicate Q, $\mu n \boldsymbol{.} Q(n)$ denotes the minimum integer n such that $Q(n)$ is true if such an n exists; it is undefined otherwise. For any set A, 2^A denotes the power set of A. A^k denotes the Cartesian product of A with itself k times. The quantifiers '$\overset{\infty}{\forall}$', '$\exists!$' and '$\overset{\infty}{\exists}$' mean 'for all but finitely many', 'there exists a unique' and 'there exist infinitely many' respectively.

3. Preliminaries.

The kind of data a scientist handles in the investigation of a phenomenon F is an ordered pair $(x, f(x))$, where f is the function associated with F and $f(x)$ is the result of experiment x on F. At any given time, a scientist conjectures a hypothesis after seeing a finite sequence of such ordered pairs. We let SEQ denote the set of all finite sequences of ordered pairs. Finite sequences are also referred to as initial segments. As already mentioned, a hypothesis is simply a computer program identified by its index in a given fixed acceptable programming system. Based on these observations, we describe a learning machine in Definition 3.1 below. We let σ and τ, with or without decorations, range over SEQ. $\mathrm{content}(\sigma)$ denotes the set of pairs appearing in σ. The length of σ, denoted by $|\sigma|$, is the number of elements in σ. $\sigma \diamond (x, y)$ denotes the *concatenation* of (x, y) at the end of sequence σ.

Definition 3.1. A *learning machine* is an algorithmic device that computes a mapping from SEQ into N. We let $\mathbf{M}$, with or without decorations, range over learning machines.

Scientific inquiry is a limiting process. There is no fixed order in which experiments may be performed, and a scientist is never sure if any new evidence would cause a revision of the currently held hypothesis. The notion of a *text* is described in Definition 3.2 to model the infinite sequence of experimental data a scientist may encounter in the course of investigating a phenomenon.

Definition 3.2.

1. A *text* is any infinite sequence of ordered pairs. We let T, with or without decorations, range over texts.
2. The set of pairs appearing in a text T is denoted by content(T).
3. Let a total function $f : N \to N$, and a text T be given. T is for f just in case content$(T) = \{(x, y) \mid f(x) = y\}$.
4. The initial finite sequence of T of length n is denoted by $T[n]$.

Definition 3.3 below describes what it means for a learning machine to converge on a text.

Definition 3.3.

Suppose $\mathbf{M}$ is a learning machine and T is a text. $\mathbf{M}(T)\!\downarrow$ (read: $\mathbf{M}(T)$ *converges*) $\iff$ $(\exists p)(\overset{\infty}{\forall} n)[\mathbf{M}(T[n]) = p]$. If $\mathbf{M}(T)\!\downarrow$, then $\mathbf{M}(T)$ is defined $=$ the unique p such that $(\overset{\infty}{\forall} n)[\mathbf{M}(T[n]) = p]$; otherwise $\mathbf{M}(T)$ is said to diverge (written: $\mathbf{M}(T)\!\uparrow$).

Definition 3.4 below describes what it means for a learning machine to successfully learn a function.

Definition 3.4. [12, 3, 8]

1. $\mathbf{M}$ $\mathbf{Ex}^a$-*identifies* a total function f (written: $f \in \mathbf{Ex}^a(\mathbf{M})$) $\iff$ ($\forall$ texts T for f)$(\exists p \mid \varphi_p =^a f)[\mathbf{M}(T)\!\downarrow = p]$.
2. $\mathbf{Ex}^a = \{\mathcal{S} \mid (\exists \mathbf{M})[\mathcal{S} \subseteq \mathbf{Ex}^a(\mathbf{M})]\}$.

Definition 3.4 above models the situation in which a scientist has access to an *accurate* source of data. Since totally accurate experimental data is seldom available, models of scientific inquiry should accommodate inaccuracies. In the paradigm under discussion, inaccurate data is modeled by inaccurate texts. First, we consider in brief the kinds of inaccuracies that may arise in experimentation. The subject of inaccuracies in the data available to a learning machine has been previously studied by Schäfer-Richter [23], Fulk and Jain [11], Osherson, Stob and Weinstein [17], Jain [14, 15].

- **Noisy Data:** Experimental error, usually caused by faulty equipment, may result in spurious data that is not representative of the phenomenon under investigation.
- **Incomplete Data:** It may not be feasible to perform certain experiments either due to technological limitations or due to ethical considerations. Such situations result in incomplete data.
- **Imperfect Data:** In most experimental investigations, the inaccuracies are a mixture of both noisy and incomplete data. Such situations are said to yield imperfect data.

The three kinds of inaccuracies discussed above suggest three natural extensions to the notion of texts defined below.

Definition 3.5. [11, 17, 14] (also see [23]) Let a text T and a function $f \in \mathcal{R}$ be given. Let $a \in N \cup \{*\}$. Then,

1. T is said to be *a-noisy* for f just in case $\{(x, y) \mid f(x) = y\} \subseteq$ content(T) and card(content$(T) - \{(x, y) \mid f(x) = y\}) \leq a$.
2. T is said to be *a-incomplete* for f just in case content$(T) \subseteq \{(x, y) \mid f(x) = y\}$ and card$(\{(x, y) \mid f(x) = y\} -$ content$(T)) \leq a$.
3. T is said to be *a-imperfect* for f just in case card$(\{(x, y) \mid f(x) = y\} \mathbf{\Delta}$content$(T)) \leq a$.

Definition 3.6 below describes what it means for a learning machine to learn a function from inaccurate texts. We give the definition for noisy texts; the corresponding notions for incomplete and imperfect texts (respectively called, $\mathbf{In}^a\mathbf{Ex}^b$ and $\mathbf{Im}^a\mathbf{Ex}^b$) can be defined similarly.

Definition 3.6. [11, 17, 14] (also see [23]) Let $a, b \in N \cup \{*\}$.

1. $\mathbf{M}$ $\mathbf{N}^a\mathbf{Ex}^b$-*identifies* a total function f (written: $f \in \mathbf{N}^a\mathbf{Ex}^b(\mathbf{M})$) $\iff$ ($\forall$ a-noisy texts T for f)$(\exists p \mid \varphi_p =^b f)[\mathbf{M}(T)\!\downarrow = p]$.
2. $\mathbf{N}^a\mathbf{Ex}^b = \{\mathcal{S} \mid (\exists \mathbf{M})[\mathcal{S} \subseteq \mathbf{N}^a\mathbf{Ex}^b(\mathbf{M})]\}$.

4. Multiple Inaccurate Texts.

The previous section described a paradigm that models a scientist receiving data from a *single*, albeit possibly inaccurate source. In actual scientific practice, a phenomenon is investigated by a number of different scientists, each performing their own experiments. In due course of time, the data of one scientist becomes available to another through scientific journals, word of mouth, personal communications, etc. Thus, a scientist conjectures hypotheses based on data coming from a number of different sources. This situation could be modeled in the present paradigm as a machine receiving multiple texts, some or all of which are inaccurate. It should be noted that the presence of inaccuracies in at least some of the texts is essential for this model to be different because learning from multiple texts, each of which are accurate, is equivalent to learning from a single accurate text.

Before we define learning from multiple inaccurate texts, we need to tinker with the definition of a learning machine to account for data coming from more than one source. Definition 4.1 below describes learning machines that receive multiple streams of data.

Definition 4.1. Let $k \in N^+$. A *learning machine with k streams* is an algorithmic device that computes a mapping from SEQ^k into N.

Again, we let $\mathbf{M}$, with or without decorations, range over learning machines with multiple streams; it will be clear from context if we mean a learning machine with a single stream. Definition 4.2 below describes what it means for a learning machine to converge on multiple texts.

Definition 4.2. Let $k \in N^+$. Let $\mathbf{M}$ be a learning machine and $T_1, T_2, \ldots, T_k$ are k texts. $\mathbf{M}(T_1, T_2, \ldots, T_k)\!\downarrow$ (read: $\mathbf{M}(T_1, T_2, \ldots, T_k)$ converges) $\iff$ $(\exists p)(\exists n)(\forall n_1, n_2, \ldots, n_k \mid n_1 \geq n, n_2 \geq n, \ldots, n_k \geq n)[\mathbf{M}(T_1[n_1], T_2[n_2], \ldots, T_k[n_k]) = p]$. If $\mathbf{M}(T_1, T_2, \ldots, T_k)\!\downarrow$, then $\mathbf{M}(T_1, T_2, \ldots, T_k)$ is defined = the unique p such that $(\exists n)(\forall n_1, n_2, \ldots, n_k \mid n_1 \geq n, n_2 \geq n, \ldots, n_k \geq n)[\mathbf{M}(T_1[n_1], T_2[n_2], \ldots, T_k[n_k]) = p]$; otherwise $\mathbf{M}(T_1, T_2, \ldots, T_k)$ is said to diverge (written: $\mathbf{M}(T_1, T_2, \ldots, T_k)\!\uparrow$).

Definition 4.3 below describes what it means for a learning machine to learn a function from multiple number of inaccurate texts. We give the definition for noisy texts; the corresponding notions for incomplete and imperfect texts may be defined similarly.

Definition 4.3. Let $j, k \in N^+$. Let $a, b \in N \cup \{*\}$.

(a) A learning machine $\mathbf{M}$ $\mathbf{Mul}^j_k\mathbf{N}^a\mathbf{Ex}^b$-*identifies* a total function f (written: $f \in \mathbf{Mul}^j_k\mathbf{N}^a\mathbf{Ex}^b(\mathbf{M})$) $\iff$ ($\forall$ k texts $T_1, T_2, \ldots, T_k$ such that at least j out of these k texts are a-noisy for f)$(\exists p \mid \varphi_p =^b f)[\mathbf{M}(T_1, T_2, \ldots, T_k)\!\downarrow = p]$.

(b) $\mathbf{Mul}^j_k\mathbf{N}^a\mathbf{Ex}^b = \{\mathcal{S} \mid (\exists \mathbf{M})[\mathcal{S} \subseteq \mathbf{Mul}^j_k\mathbf{N}^a\mathbf{Ex}^b(\mathbf{M})]\}$.

So, a machine $\mathbf{Mul}^j_k\mathbf{N}^a\mathbf{Ex}^b$- ($\mathbf{Mul}^j_k\mathbf{In}^a\mathbf{Ex}^b$-, $\mathbf{Mul}^j_k\mathbf{Im}^a\mathbf{Ex}^b$-) identifying a function f, upon being fed k texts at least j of which are a-noisy (a-incomplete, a-imperfect) for f, converges to a program for a b-variant for f. Henceforth, when discussing $\mathbf{Mul}^j_k\mathbf{N}^a\mathbf{Ex}^b$, etc., those texts for which the inaccuracy is within the required bound are referred to as *acceptable* texts. We next note that the only interesting cases are those for which the number of acceptable texts is a majority of the total number of texts.

Consider a collection of recursive functions $\mathcal{C}$. Assume $f_1, f_2 \in \mathcal{C}$ to be such that $f_1 \neq^{2a} f_2$. Then, for all k, we have $\mathcal{C} \notin \mathbf{Mul}^{\lfloor \frac{k}{2} \rfloor}_k\mathbf{N}^0\mathbf{Ex}^a$. This is because among k texts there may be $\lfloor \frac{k}{2} \rfloor$ accurate texts for both f_1 and f_2. To avoid such problems, we only consider those cases in which a strict majority of the texts are acceptable. [2]

We now discuss our results before presenting them with proofs in the next section.

One of our central aims is to investigate the relationship between identification from a single inaccurate text and identification from multiple inaccurate texts, where the inaccuracy is of the same kind. For noisy texts, for instance, this question can be phrased as: How does $\mathbf{N}^a\mathbf{Ex}^b$ relate to $\mathbf{Mul}^j_k\mathbf{N}^c\mathbf{Ex}^d$ for various values of a, b, c, d and j, k? Similar questions can be posed for incomplete and imperfect texts. This general question turns out to be combinatorially very difficult, and only partial results are presented in this paper.

First, the following immediate proposition states that for the same bound on the number of inaccuracies and the same kind of inaccuracy, identification from multiple inaccurate texts cannot be better than identification from a single inaccurate text, that is, collections of functions that can be identified from multiple inaccurate texts can also identified from a single inaccurate text.

Proposition 4.4. *Let $j, k \in N$, such that $k \geq j$. Let $a, b \in N \cup \{*\}$.*

1. $\mathbf{Mul}^j_k\mathbf{N}^a\mathbf{Ex}^b \subseteq \mathbf{N}^a\mathbf{Ex}^b$.
2. $\mathbf{Mul}^j_k\mathbf{In}^a\mathbf{Ex}^b \subseteq \mathbf{In}^a\mathbf{Ex}^b$.
3. $\mathbf{Mul}^j_k\mathbf{Im}^a\mathbf{Ex}^b \subseteq \mathbf{Im}^a\mathbf{Ex}^b$.

The natural question is when does the above proposition yield an equality. That is, we would like to have sufficient conditions for when identification from multiple texts is no worse than identification from a single inaccurate text. Results in Section 5.1 give some conditions such that,— for each kind of inaccuracy (noise, incompleteness, and imperfection)—if a collection of functions can be learned from a single inaccurate text, then it can also be learned (with the same bound on the number of errors in the final program) from multiple inaccurate texts, provided the bound on the number of inaccuracies in at least a *majority* of the acceptable texts is the same as the bound on the number of inaccuracies in the single inaccurate text.

The next natural question is under what conditions identification from multiple texts, some of which may be inaccurate, constitutes a restriction on identification from a single inaccurate text. More precisely, we would like to answer the question: For what relationship between a, b, c, d and j, k, $\mathbf{N}^a\mathbf{Ex}^b - \mathbf{Mul}^j_k\mathbf{N}^c\mathbf{Ex}^d \neq \emptyset$? Unfortunately, this question turns out to be very difficult, and we are only able to provide partial results for identification from 3 texts, at least 2 of which are acceptable (that is, for the case $j = 2$ and $k = 3$). Theorem 5.4 in Section 5.2 is a very general result that gives sufficient conditions for a, b, c, d such that $\mathbf{N}^a\mathbf{Ex}^b - \mathbf{Mul}^2_3\mathbf{N}^c\mathbf{Ex}^d \neq \emptyset$. Unfortunately these conditions turn out to be somewhat intricate, but the reader can get a feel for special cases of these conditions presented in Corollaries 5.5, 5.6, 5.7,

[2] It should be noted that if for all $f_1, f_2 \in \mathcal{C}$, $f_1 =^{2a} f_2$, then all functions in $\mathcal{C}$ are finite variants of any fixed function in the class. We feel that such cases are not interesting, because if at least j out of k input texts are acceptable for a function $f \in \mathcal{C}$, then one can easily find a program p (in the limit) such that, at least j out of k input texts are acceptable for φ_p (though, this φ_p may not be in $\mathcal{C}$).

and 5.8. These results are presented in Section 5.2.

Also, in Section 5.2, we consider the cases where the restrictive effects of learning from multiple inaccurate texts can be compensated. More precisely, we identify the relationship between a, b, c, d such that $\mathbf{N}^a\mathbf{Ex}^b \subseteq \mathbf{Mul}_3^2\mathbf{N}^c\mathbf{Ex}^d$. This is achieved by either allowing a greater upper bound on the number of errors tolerated in the final program (that is, by making d greater than b) or by making the quality of acceptable texts better than the quality of single inaccurate text (that is, by making c smaller than a). An interesting result on these lines is that for each type of inaccuracy, the collections of functions identified from a single inaccurate text can also be identified from 3 texts, at least 2 of which are acceptable with the same bound on the number of inaccuracies as the single text, provided we are prepared to tolerate twice the number of errors in the final program. This follows from Theorems 5.9, 5.19, and 5.22.

In Section 5.3, we consider the effects of increasing the bound on the number of errors tolerated in the final program keeping other parameters the same and the effects of increasing the bound on the number of inaccuracies in the acceptable texts keeping other parameters the same. We are able to show the expected result that for each inaccuracy type, larger collections of functions become identifiable if we

- increase the bound on the number of errors allowed in the final program (keeping other parameters the same), or
- decrease the bound on the number of inaccuracies allowed in the input texts (keeping other parameters the same).

Results discussed so far have been about the same kind of inaccuracies. Finally, in Section 5.4, we present results that compare one kind of inaccuracy with another in the context of identification from multiple texts. Our findings show that noisy texts are better for learnability than incomplete texts. This observation may be interpreted as saying that spurious data is preferable to missing data. However, we also show that imperfect texts are worse for learnability than incomplete texts, which can be seen as saying that a mixture of spurious and missing data is less desirable than only missing data.

5. Results.

We now present results relating various criteria of inference introduced in Section 4. Most results in the present paper are about the case in which a learning machine is receiving data from 3 sources, at least 2 of which are acceptable. Section 5.1 presents results about those cases in which learning from multiple texts is not a restriction on learning power. Section 5.2 presents results where learning from multiple texts is a restriction on learning power, and also about cases where the deleterious effects of learning from multiple inaccurate texts can be compensated by either allowing extra errors in the final program or by improving the quality of the acceptable texts. Section 5.3 presents the hierarchy results and Section 5.4 presents some results about the interaction between different kinds of of inaccuracies.

From a purely technical point of view, results presented in Section 5.2 are the most difficult results of this paper. Also, we are able to establish more results for noisy texts than for incomplete and imperfect texts.

5.1. When Identification from Multiple Texts is Not Restrictive.

We consider cases when learning from multiple inaccurate texts is equivalent to learning from a single inaccurate text. Parts (1), (2), and (3) of the next result show that for each kind of inaccuracy, the collections of functions that can be identified from a single text with a finite number of inaccuracies is exactly the same as the collections of functions that can be identified (with the same bound on the number of errors allowed in the final program) from multiple texts, at least a majority of which have only a finite number of inaccuracies.

Theorem 5.1. *Let $j, k \in N$, $k \geq j > \lfloor \frac{k}{2} \rfloor$. Let $a \in N \cup \{*\}$.*

1. $\mathbf{Mul}^j_k\mathbf{N}^*\mathbf{Ex}^a = \mathbf{N}^*\mathbf{Ex}^a$.
2. $\mathbf{Mul}^j_k\mathbf{In}^*\mathbf{Ex}^a = \mathbf{In}^*\mathbf{Ex}^a$.
3. $\mathbf{Mul}^j_k\mathbf{Im}^*\mathbf{Ex}^a = \mathbf{Im}^*\mathbf{Ex}^a$.
4. $\mathbf{Mul}^j_k\mathbf{N}^0\mathbf{Ex}^a = \mathbf{Mul}^j_k\mathbf{In}^0\mathbf{Ex}^a = \mathbf{Mul}^j_k\mathbf{Im}^0\mathbf{Ex}^a = \mathbf{Ex}^a$.

Proof. Given k texts $T_1, T_2, \ldots, T_k$, construct a text T such that $(x, y) \in \text{content}(T) \iff \text{card}(\{i \mid (x, y) \in \text{content}(T_i)\}) \geq \lfloor \frac{k}{2} \rfloor + 1$. It is easy to see that if at least $\lfloor \frac{k}{2} \rfloor + 1$ of the k texts are $*$-noisy ($*$-incomplete, $*$-imperfect, accurate) for f then so is T. The theorem follows. □

We now introduce a technical notion that is used in later proofs. Let P be a finite set of programs. Define program $\mathbf{Unify}(P)$ as follows:

```
begin $\varphi_{\mathbf{Unify}(P)}(x)$
    Search for $i \in P$ such that $\varphi_i(x)\downarrow$.
    If and when such an $i$ is found, output $\varphi_i(x)$ (for the first such $i$ found).
end $\varphi_{\mathbf{Unify}(P)}(x)$
```

The next result shows that for $a \in N \cup \{*\}$, the collections of functions for which an exact program can be identified from a single text with the number of inaccuracies bounded by a is exactly the same as the collections of functions for which an exact program can be identified from multiple texts, at least a majority of which have at most a inaccuracies.

Theorem 5.2. *Let $j, k \in N$, $k \geq j > \lfloor \frac{k}{2} \rfloor$. Let $a \in N \cup \{*\}$.*

1. $\mathbf{Mul}^j_k\mathbf{N}^a\mathbf{Ex} = \mathbf{N}^a\mathbf{Ex}$.
2. $\mathbf{Mul}^j_k\mathbf{In}^a\mathbf{Ex} = \mathbf{In}^a\mathbf{Ex}$.
3. $\mathbf{Mul}^j_k\mathbf{Im}^a\mathbf{Ex} = \mathbf{Im}^a\mathbf{Ex}$.

Proof. (1) Proposition 4.4 yields $\mathbf{Mul}^j_k\mathbf{N}^a\mathbf{Ex} \subseteq \mathbf{N}^a\mathbf{Ex}$. We now show that $\mathbf{N}^a\mathbf{Ex} \subseteq \mathbf{Mul}^j_k\mathbf{N}^a\mathbf{Ex}$.

Let $\mathbf{M}$ $\mathbf{N}^a\mathbf{Ex}$-identify $\mathcal{C}$. We show how to $\mathbf{Ex}$-identify $f \in \mathcal{C}$ from k texts at least j of which are a-noisy for f. Let n, $S = \{i_1, i_2, \ldots, i_j\} \subseteq \{1, 2, \ldots, k\}, P = \{p_{i_1}, p_{i_2}, \ldots, p_{i_j}\}$, be such that,

(i) for all $l \in S$ and $m \geq n$, $\mathbf{M}(T_l[m]) = p_l$,
(ii) if $p, q \in P$, then $\text{card}(\{x \mid \varphi_p(x)\downarrow \neq \varphi_q(x)\downarrow\}) = 0$, and
(iii) $\text{card}(S) = j$.

Clearly, such n, S, P exist (let $S \subseteq \{i \mid T_i$ is a-noisy for $f\}$ be a set of cardinality j; let n be such that, for all $l \in S$, $\mathbf{M}$ converges on T_l after seeing at most n inputs; let $P = \{\mathbf{M}(T_l) \mid l \in S\}$. Then n, S, P satisfy (i), (ii), and (iii)). Clearly, a machine $\mathbf{M}'$, given texts $T_1, \ldots, T_k$, can find such an n, S, P in the limit. Now since $\mathbf{M}$ on $T_l, l \in S$, converges to p_l and $\text{card}(S) > k - j$, there exists an $l \in S$ such that $\varphi_{p_l} = f$. This along with (ii) above implies that $\mathbf{Unify}(P)$ is a program for f. This proves Part (1).

Part (2) and (3) can be proved similarly. □

The next result shows that for $a \in N \cup \{*\}$, the collections of functions for which a $*$-error program can be identified from a single text with the number of inaccuracies bounded by a is exactly the same as the collections of functions for which a $*$-error program can be identified from multiple texts, at least a majority of which have at most a inaccuracies.

Theorem 5.3. *Let $j, k \in N$, $k \geq j > \lfloor \frac{k}{2} \rfloor$. Let $a \in N \cup \{*\}$.*

1. $\mathbf{Mul}^j_k\mathbf{N}^a\mathbf{Ex}^* = \mathbf{N}^a\mathbf{Ex}^*$.

2. $\mathbf{Mul}^j_k\mathbf{In}^a\mathbf{Ex}^* = \mathbf{In}^a\mathbf{Ex}^*$.

3. $\mathbf{Mul}^j_k\mathbf{Im}^a\mathbf{Ex}^* = \mathbf{Im}^a\mathbf{Ex}^*$.

Proof. (1) Proposition 4.4 yields $\mathbf{Mul}^j_k\mathbf{N}^a\mathbf{Ex}^* \subseteq \mathbf{N}^a\mathbf{Ex}^*$. We now show that $\mathbf{N}^a\mathbf{Ex}^* \subseteq \mathbf{Mul}^j_k\mathbf{N}^a\mathbf{Ex}^*$.

Let $\mathbf{M}$ $\mathbf{N}^a\mathbf{Ex}^*$-identify $\mathcal{C}$. We show how to $\mathbf{Ex}^*$-identify $f \in \mathcal{C}$ from k texts at least j of which are a-noisy for f. Let $n, S = \{i_1, i_2, \ldots, i_j\} \subseteq \{1, 2, \ldots, k\}, P = \{p_{i_1}, p_{i_2}, \ldots, p_{i_j}\}$, be such that,
(i) for all $l \in S$ and $m \geq n$, $\mathbf{M}(T_l[m]) = p_l$,
(ii) if $p, q \in P$, then $\mathrm{card}(\{x \mid \varphi_p(x)\downarrow \neq \varphi_q(x)\downarrow\}) \leq n$, and
(iii) $\mathrm{card}(S) = j$.
Clearly, such n, S, P exist (let $S \subseteq \{i \mid T_i$ is a-noisy for $f\}$ be a set of cardinality j; let $P = \{\mathbf{M}(T_l) \mid l \in S\}$; let n be so large that for all $l \in S$, $\mathbf{M}$ converges on T_l after seeing at most n inputs and $f =^{n/2} \varphi_p$, for all $p \in P$; Then n, S, P satisfy (i), (ii) and (iii)). Clearly, a machine $\mathbf{M}'$, given texts $T_1, \ldots, T_k$, can find such an n, S, P in the limit. Now since $\mathbf{M}$ on $T_l, l \in S$, converges to p_l and $\mathrm{card}(S) > k - j$, there exists an $l \in S$ such that $\varphi_{p_l} =^* f$. This along with (ii) above implies that $\mathbf{Unify}(P)$ is a program for a finite variant of f. This proves Part (1).

Parts (2) and (3) can be proved similarly. □

5.2. When Identification from Multiple Texts is Restrictive. In this section we initially tackle the question of when, for the same kind of inaccuracy, learning from multiple texts, some of which may be inaccurate, is a restriction over learning from a single inaccurate text. That is, we consider the question: For which values of a, b, c, d and j, k, is $\mathbf{N}^a\mathbf{Ex}^b - \mathbf{Mul}^j_k\mathbf{N}^c\mathbf{Ex}^d \neq \emptyset$? As already noted this turns out to be very difficult question. For the case $j = 2$ and $k = 3$, we are able to provide significant partial answers. These results follow from a very complex diagonalization argument.

In this section, we then consider the related question of when the restriction of identification from multiple texts can be compensated. That is, for what values of a, b, c, d, is $\mathbf{N}^a\mathbf{Ex}^b \subseteq \mathbf{Mul}^2_3\mathbf{N}^c\mathbf{Ex}^d$. These results use simulation arguments.

We first present results for noisy texts (Section 5.2.1), followed by results for incomplete texts (Section 5.2.2). Only few results are presented for imperfect texts.

5.2.1. Noisy Texts. We present a general theorem that covers numerous cases in which identification from multiple noisy texts is a restriction over identification from a single noisy text. Unfortunately, the conditions in the theorem turn out to be very complex, and the reader is advised to see the four corollaries, presented immediately after the statement of the theorem, for cases in which multiple texts pose a restriction. However, we first try to give a intuitive motivation of the theorem. Consider the question of where we are trying to compare identification from a single noisy text and identification from 3 texts, at least 2 of which are noisy. Further let,

- a be the bound on the noise level of the single text,
- b be the bound on the number of errors allowed in programs inferred from the single text,
- c be the bound on the noise level of the acceptable texts in the case of identification from multiple texts, and
- d be the bound on the number of errors allowed in programs inferred from multiple texts.

We would like to know when

$$\mathbf{N}^a\mathbf{Ex}^b - \mathbf{Mul}^2_3\mathbf{N}^c\mathbf{Ex}^d \neq \emptyset.$$

Now, let us see the effects of altering each of the parameters. Increasing a or decreasing b makes the class $\mathbf{N}^a\mathbf{Ex}^b$ smaller. Similarly, decreasing c and increasing d makes the class $\mathbf{Mul}^2_3\mathbf{N}^c\mathbf{Ex}^d$ larger. The following theorem tells us how much we can "stretch" the parameters a, b, c, d such that there are still collections of functions that can be identified from single noisy text but not from 3 texts, at least 2 of which are acceptable.

Theorem 5.4. *Let $a, b, c, d \in N$ be given. If there exist $r, \alpha \in N$ such that,*
$r - b \leq \alpha \leq \min(\{b, r\})$,
$r \leq c$,
$2r > a$, *and*
$d < \max(\{b + \alpha - \frac{r}{2}, b + \frac{\alpha}{3}\})$,
then $\mathbf{N}^a\mathbf{Ex}^b - \mathbf{Mul}^2_3\mathbf{N}^c\mathbf{Ex}^d \neq \emptyset$. Before giving a proof of the above result, we present corollaries. The first corollary says that for given $a, b, c \in N$ such that a is greater, but not too greater, than c ($c \leq a \leq 2c - 1$) and c is no greater than b, there are collections of functions for which a b-error program can be identified from an a-error text, but for which even a $(b + \lceil\frac{c}{2}\rceil - 1)$-error program cannot be identified from 3 texts, at least 2 of which are c-noisy for the function being learned.

Corollary 5.5.
Let $a, b, c \in N$ be such that $c \leq a \leq 2c - 1$, $c \leq b$. Then, $\mathbf{N}^a\mathbf{Ex}^b - \mathbf{Mul}^2_3\mathbf{N}^c\mathbf{Ex}^{b+\lceil\frac{c}{2}\rceil-1} \neq \emptyset$.

Proof. Take $r = \alpha = c$ in Theorem 5.4. □

The next three corollaries are more variations on this theme and give insight into how much the parameters a, b, c, d can be stretched.

Corollary 5.6. *Let $a, b, c \in N$ be such that $c \leq a \leq 2c - 1$, $\frac{c}{2} \leq b \leq \frac{3}{4}\lceil\frac{a+1}{2}\rceil$. Then,* $\mathbf{N}^a\mathbf{Ex}^b - \mathbf{Mul}^2_3\mathbf{N}^c\mathbf{Ex}^{\lceil\frac{4b}{3}\rceil-1} \neq \emptyset$.

Proof. Take $r = c, \alpha = b$ in Theorem 5.4. □

Corollary 5.7. *Let $a, b, c \in N$ be such that $c \leq a \leq 2c - 1$, $\frac{a}{2} < b \leq c$. Then,* $\mathbf{N}^a\mathbf{Ex}^b - \mathbf{Mul}^2_3\mathbf{N}^c\mathbf{Ex}^{\lceil\frac{3b}{2}\rceil-1} \neq \emptyset$.

Proof. Take $r = \alpha = b$ in Theorem 5.4. □

Corollary 5.8. *Let $a, b, c \in N$ be such that $c \leq a \leq 2c-1$, $\max(\{\frac{c}{2}, \frac{3}{4}\lceil\frac{a+1}{2}\rceil\}) \leq b \leq \frac{a}{2}$. Then* $\mathbf{N}^a\mathbf{Ex}^b - \mathbf{Mul}^2_3\mathbf{N}^c\mathbf{Ex}^{\lceil 2b-\frac{a+2}{4}\rceil-1} \neq \emptyset$.

Proof. Take $r = \lceil\frac{a+1}{2}\rceil, \alpha = b$ in Theorem 5.4. □

We now give a proof of Theorem 5.4. The proof turns out to be somewhat complex because we have to come up with a collection of functions that can be identified from a single noisy text, but which cannot be identified from 3 texts, at least 2 of which are noisy. To design this collection of functions, we employ a technique from the study of identification from single inaccurate texts [11]. We then employ the operator recursion theorem [5] to construct a diagonalization argument to show that this class cannot be identified from 3 texts, at least 2 of which are noisy. The diagonalization technique is an adaptation of the techniques used elsewhere in the study of inductive inference; we direct the reader to a survey of such techniques [6].

Proof. (Theorem 5.4) Assume the hypothesis and fix r, α satisfying the hypothesis.

Let $A = \{\langle 0, x\rangle \mid 0 \leq x < r\}$.

Consider the following class of functions.

$\mathcal{C} = \{f \in \mathcal{R} \mid$ the following conditions hold:
(1) $f(A) = f(\langle 0, 0\rangle) \in \{1, 2, 3\}$;
(2) $(\forall i \in \{1, 2, 3\} - \{f(\langle 0, 0\rangle)\})[(\max(\{f(\langle i, \langle x, 0\rangle\rangle) \mid x \in N\})$ exists$) \wedge$
$(\varphi_{\max(\{f(\langle i,\langle x,0\rangle\rangle)\mid x\in N\})} =^b f)]$;
(3) $(\forall i \in \{1, 2, 3\})(\forall x, y, z)[f(\langle i, \langle x, y\rangle\rangle) = f(\langle i, \langle x, z\rangle\rangle)]$ $\}$

Intuitively, A is the set of points where some coding is done. (A is the only place where inaccuracies will matter). a-noise is not enough to spoil the coding in A for $\mathbf{N}^a\mathbf{Ex}^b$-identification. However, r-noise is enough to spoil the coding for $\mathbf{Mul}_3^2\mathbf{N}^c\mathbf{Ex}^d$-identification.

It is easy to show that $\mathcal{C} \in \mathbf{N}^a\mathbf{Ex}^b$. To see this suppose T is a-noisy text for $f \in \mathcal{C}$. Since $2r > a$, there exists an $i \in \{1,2,3\}$ and $x \in A$, such that $(x,i) \notin \text{content}(T)$. This along with clause (2) in the definition of $\mathcal{C}$ implies that $\varphi_{\max(\{f(\langle i,\langle x,0\rangle\rangle) \mid x \in N\})} =^b f$. Now, $\max(\{f(\langle i,\langle x,0\rangle\rangle) \mid x \in N\})$, can be determined, in the limit, from T due to cylindrification in clause (3). It follows that $\mathcal{C} \in \mathbf{N}^a\mathbf{Ex}^b$.

We now show that $\mathcal{C} \notin \mathbf{Mul}_3^2\mathbf{N}^c\mathbf{Ex}^d$.

The essentially idea for diagonalization is that, a machine cannot $\mathbf{Ex}^d$-identify $\mathcal{C}$ if the coding in A (due to clause (1) in definition of $\mathcal{C}$) is spoiled. So, suppose by way of contradiction that machine $\mathbf{M}$ $\mathbf{Mul}_3^2\mathbf{N}^c\mathbf{Ex}^d$-identifies $\mathcal{C}$.

Then by the *Operator Recursion Theorem*[5], there exists a recursive, 1-1 and increasing p, with $p(0) > 1$, defined as follows in stages.

We define $\varphi_{p(i)}$ in Stages $s \geq 3$. Let x_3^s denote the least x such that, for all y, $\varphi_{p(1)}(\langle 3,\langle x,y\rangle\rangle)$ is not defined before Stage s.
For $x < \alpha$, let $\varphi_{p(1)}(\langle 0,x\rangle) = 2$. For $\alpha \leq x < r$, let $\varphi_{p(1)}(\langle 0,x\rangle) = 3$. Let $\varphi_{p(1)}(\langle 1,\langle 0,0\rangle\rangle) = p(1)$, and $\varphi_{p(1)}(\langle 2,\langle 0,0\rangle\rangle) = p(2)$.
For $x < \alpha$, let $\varphi_{p(2)}(\langle 0,x\rangle) = 1$. For $\alpha \leq x < r$, let $\varphi_{p(2)}(\langle 0,x\rangle) = 3$. Let $\varphi_{p(2)}(\langle 1,\langle 0,0\rangle\rangle) = p(1)$, and $\varphi_{p(2)}(\langle 2,\langle 0,0\rangle\rangle) = p(2)$.
Let σ_1^3, σ_2^3 and σ_3^3 be such that
$\text{content}(\sigma_1^3) = \{(x,2) \mid x \in A\} \cup \{(x,3) \mid x \in A\} \cup \{(\langle 1,\langle 0,0\rangle\rangle, p(1)), (\langle 2,\langle 0,0\rangle\rangle, p(2))\}$,
$\text{content}(\sigma_2^3) = \{(x,1) \mid x \in A\} \cup \{(x,3) \mid x \in A\} \cup \{(\langle 1,\langle 0,0\rangle\rangle, p(1)), (\langle 2,\langle 0,0\rangle\rangle, p(2))\}$,
$\text{content}(\sigma_3^3) = \{(x,1) \mid x \in A\} \cup \{(x,2) \mid x \in A\} \cup \{(\langle 1,\langle 0,0\rangle\rangle, p(1)), (\langle 2,\langle 0,0\rangle\rangle, p(2))\}$.

Intuitively, we will form three (nearly identical) texts T_1, T_2, and T_3 with the initial segments $\sigma_1^3, \sigma_2^3, \sigma_3^3$ as defined above. Note that these three texts spoil the coding in A (since all possible values of the coding are present in two of these texts). Now the basic idea in the diagonalization below, is to either

(1) force infinitely many mind changes by $\mathbf{M}$ (see step 4 in the construction) — in this case T_1 and T_2 form the c-noisy texts for the diagonalizing function — or

(2) based on what the final program $\mathbf{M}(T_1,T_2,T_3)$, does on the inputs from A (see step 2 in the construction), we do an appropriate diagonalization of the same form as done in $\mathbf{Ex}$-hierarchy theorem by [8] (see steps 5a,5c and 6a in the construction). The diagonalization is done so as to maximize the errors of the final program under the constraint of keeping the diagonalizing function within the class (see the comments at the end of step 2 in the construction). Steps 5b and 6b in the construction are needed since one cannot effectively determine whether program $\mathbf{M}(T_1,T_2,T_3)$ halts on inputs from the set A (we can do it only in the limit).
Let $\varphi_{p(i)}^s$ denote the part of $\varphi_{p(i)}$ defined before Stage s. Go to Stage 3.

Begin Stage s

1. Let $q = \mathbf{M}(\sigma_1^s, \sigma_2^s, \sigma_3^s)$.

2. Let $D_1 = \{x \mid x \in A \ \wedge \ \Phi_q(x) \leq s \ \wedge \ \varphi_q(x) = 1\}$.
Let $D_2 = \{x \mid x \in A \ \wedge \ \Phi_q(x) \leq s \ \wedge \ \varphi_q(x) = 2\}$.
Let $D_3 = \{x \mid x \in A \ \wedge \ \Phi_q(x) \leq s \ \wedge \ \varphi_q(x) \in N - \{1,2\}\}$.
Let $err_{1,2} = r - \text{card}(D_3) + b - \alpha$.
Let $err_{2,3} = r - \text{card}(D_1) + b - r + \alpha$.

Let $err_{1,3} = r - \mathrm{card}(D_2) + b - r + \alpha$.
If $err_{1,2} \leq \max(\{err_{1,3}, err_{2,3}\})$, then let $extra\_err = b - r + \alpha$; otherwise let $extra\_err = b - \alpha$. If $err_{1,3} > err_{2,3}$, then let $l = 1$ and $l' = 2$; otherwise let $l = 2$ and $l' = 1$.
Let $X^s \subset \{\langle 4, x\rangle \mid x \geq s\}$ denote the (lexicographically least) set of $extra\_err$ elements such that, for all $x \in X^s$, $\varphi_{p(1)}(x)$ is not defined before Stage s.
(Note: Let $T_i = \bigcup_s \sigma_i^s$. D_1 (D_2, D_3) denote the points in A for which φ_q seems to be correct if T_2 and T_3 (T_1 and T_3, T_1 and T_2) are the acceptable texts for the function being learned and $err_{i,j}$ denotes the error we can force if we chose a function such that T_i, T_j are correct texts for the function. Informally, an attempt is made at every stage to try and make those two texts correct so as to maximize the number of errors made by **M**'s current output program.)

3. For $x \in A$, let $\varphi_{p(s)}(x) = l'$; Let $\varphi_{p(s)}(\langle 3, \langle x_3^s, 0\rangle\rangle) = \varphi_{p(1)}(\langle 3, \langle x_3^s, 0\rangle\rangle) = \varphi_{p(2)}(\langle 3, \langle x_3^s, 0\rangle\rangle) = p(s)$. For $x \in \mathrm{domain}(\varphi^s_{p(1)}) - A$, let $\varphi_{p(s)}(x) = \varphi_{p(1)}(x)$.

4. For $i \in \{1, 2, 3\}$, let σ_i' be an extension of σ_i^s such that $\mathrm{content}(\sigma_i') = \mathrm{content}(\sigma_i^s) \cup \{(\langle 3, \langle x_3^s, 0\rangle\rangle, p(s))\}$.
5. Dovetail Steps 5a, 5b and 5c until (if ever) one of Steps 5a, 5b succeeds and at least one iteration of the loop at Step 5c is completed. If Step 5a succeeds (before Step 5b (if ever) succeeds), then complete the then current iteration of the repeat loop in Step 5c and go to Step 6a. If Step 5b succeeds (before Step 5a (if ever) succeeds), then complete the then current iteration of the repeat loop in Step 5c and go to Step 6b.

5a. Search for $\sigma_i'' \supseteq \sigma_i'$, $i \in \{1, 2, 3\}$ such that $\mathbf{M}(\sigma_1^s, \sigma_2^s, \sigma_3^s) \neq \mathbf{M}(\sigma_1'', \sigma_2'', \sigma_3'')$ and for all $j, j' \in \{1, 2, 3\}$, for all $x, y, z \in N$, following three conditions hold:

- $\mathrm{content}(\sigma_j'') - \mathrm{content}(\sigma_j') = \mathrm{content}(\sigma_{j'}'') - \mathrm{content}(\sigma_{j'}')$
- If $(\langle j, \langle x, y\rangle\rangle, v) \in \mathrm{content}(\sigma_1'')$ and $(\langle j, \langle x, z\rangle\rangle, w) \in \mathrm{content}(\sigma_1'')$, then $v = w$.
- If $(x, y) \in (\mathrm{content}(\sigma_1'') - \mathrm{content}(\sigma_1'))$, then $[[y = 0 \ \wedge \ x \notin A]$ OR $x \in X^s$ OR [For some $i \in \{1, 2, 3\}$, $u, v \in N$: $x = \langle i, \langle u, v\rangle\rangle$ and $\varphi_{p(1)}(\langle i, \langle u, 0\rangle\rangle)$ has been defined till now and $\varphi_{p(1)}(\langle i, \langle u, 0\rangle\rangle) = y]]$.

(Remark: This substep looks for one set of suitable extensions of the currently defined initial segments of the three texts which makes **M** change its mind.)

5b. Search for $x \in A - (D_1 \cup D_2 \cup D_3)$ such that $\varphi_q(x)\downarrow$.

5c. Let $Addednew = \emptyset$.
(Note: We use $Addednew$ in order to identify the set of elements for which we extend $\varphi_{p(1)}$ in Step 5c. $Addednew$ will be used (in Steps 6a and 6b) provided at least one of Steps 5a or 5b succeeds).
Repeat

Let x be the least point not in X^s such that $\varphi_{p(1)}(x)$ is not defined till now. If for some $i \in \{1, 2, 3\}$, x is of the form $\langle i, \langle u, v\rangle\rangle$ and $\varphi_{p(1)}(\langle i, \langle u, 0\rangle\rangle)$ is defined till now, then let $\varphi_{p(1)}(x) = \varphi_{p(2)}(x) =$

$\varphi_{p(s)}(x) = \varphi_{p(1)}(\langle i, \langle u, 0\rangle\rangle)$; else, let $\varphi_{p(1)}(x) = \varphi_{p(2)}(x) = \varphi_{p(s)}(x) = 0$.

Let $Addednew = Addednew \cup \{x\}$.

Forever

6a. For all x, y such that $(x, y) \in \text{content}(\sigma_1'') - \text{content}(\sigma_1^s)$, let $\varphi_{p(1)}(x) = \varphi_{p(2)}(x) = y$. Let σ_i^{s+1} be an extension of σ_i'' such that $\text{content}(\sigma_i^{s+1}) - \text{content}(\sigma_i'') = \{(x, \varphi_{p(1)}(x)) \mid x \in Addednew\}$. Go to Stage $s+1$.

6b. Let σ_i^{s+1} be an extension of σ_i' such that $\text{content}(\sigma_i^{s+1}) - \text{content}(\sigma_i') = \{(x, \varphi_{p(1)}(x)) \mid x \in Addednew\}$. Go to Stage $s+1$.

End Stage s

Now consider the following cases:

Case 1: All stages terminate.

Let

$$f(x) = \begin{cases} 3, & \text{if } x \in A; \\ \varphi_{p(1)}(x), & \text{otherwise.} \end{cases}$$

Clearly, $f \in \mathcal{C}$ (since $\alpha \leq b$, $\varphi_{p(1)} =^b f$ and $\varphi_{p(2)} =^b f$). Also, $\mathbf{M}$ on f makes infinitely many mind changes, since Step 5b can succeed at most finitely many times before Step 5a succeeds.

Case 2: Stage s (≥ 3) is the least stage which starts but never terminates.

Let $q, err_{1,2}, err_{2,3}, err_{1,3}, D_1, D_2, D_3$ be as defined in Steps 1, 2 in Stage s. Now, $\max(\{err_{1,2},\ err_{2,3},\ err_{1,3}\}) \geq \frac{err_{1,2}+err_{2,3}+err_{1,3}}{3} \geq \frac{3b+\alpha}{3}$. Thus, $\max(\{err_{1,2},\ err_{2,3},\ err_{1,3}\}) \geq b + \frac{\alpha}{3}$. Also, since $err_{1,2} \leq r + b - \alpha$, we have $\max(\{err_{1,3}, err_{2,3}\}) \geq \frac{2b-r+2\alpha}{2}$. Thus, $\max(\{err_{1,2},\ err_{2,3},\ err_{1,3}\}) \geq \max(\{b + \alpha - \frac{r}{2}, b + \frac{\alpha}{3}\})$.

Case 2a: In Step 2 of Stage s it was found that $err_{1,2} \leq \max(\{err_{1,3}, err_{2,3}\})$.

Let l, l' be as found in Step 2. In this case $\text{domain}(\varphi_{p(s)}) = N - X^s$. Let f be any fixed total extension of $\varphi_{p(s)}$ such that, for all $x \in X^s$, we have $f(x) \neq \varphi_q(x)$. Clearly, $f \in \mathcal{C}$ (since $\varphi_{p(l)} =^b f$ and $\varphi_{p(s)} =^b f$ and $f(\langle 0, 0\rangle) = l'$). Let T_1, T_2, T_3 be extensions of $\sigma_1', \sigma_2', \sigma_3'$ (as defined in Step 4 in Stage s) respectively such that $\text{content}(T_1) - \text{content}(\sigma_1^s) = \{(x, y) \mid f(x) = y\ \wedge\ (\forall z)[(x, z) \notin \text{content}(\sigma_1^s)]\}$ and, for all j, $[|\sigma_1'| + j]^{th}$ element of T_1, $[|\sigma_2'| + j]^{th}$ element of T_2 and $[|\sigma_3'| + j]^{th}$ element of T_3 are all the same. Clearly, $\mathbf{M}(T_1, T_2, T_3) = q$ (otherwise Step 5a would succeed).

Now, for all $x \in A - D_{l'}$, $f(x) \neq \varphi_q(x)$. (Since that was the way $D_{l'}$ was chosen in Step 2 of Stage s). Thus, $\varphi_q \neq^{\max(\{err_{1,3}, err_{2,3}\})-1} f$.

Case 2b: In Step 2 of Stage s it was found that $err_{1,2} > \max(\{err_{1,3}, err_{2,3}\})$.

In this case $\text{domain}(\varphi_{p(1)}) = N - X^s$. Let f be a fixed total function such that the following three conditions hold:

- $f(A) = 3$,
- $f(x) = \varphi_{p(1)}(x)$, for $x \in \text{domain}(\varphi_{p(1)}) - A$, and
- $(\forall x \in X^s)[f(x) \neq \varphi_q(x)]$.

Clearly, $f \in \mathcal{C}$ (since $\varphi_{p(1)} =^b f$, $\varphi_{p(2)} =^b f$ and $f(\langle 0, 0\rangle) = 3$). Let T_1, T_2, T_3 be extensions of $\sigma_1', \sigma_2', \sigma_3'$ respectively such that $\text{content}(T_1) - \text{content}(\sigma_1^s) = \{(x, y) \mid f(x) = y\ \wedge\ (\forall z)[(x, z) \notin \text{content}(\sigma_1^s)]\}$ and for all j, $[|\sigma_1'| + j]^{th}$ element of T_1, $[|\sigma_2'|+j]^{th}$ element of T_2 and $[|\sigma_3'|+j]^{th}$ element of T_3 are all the same. Also, it is clear that $\mathbf{M}(T_1, T_2, T_3) = q$ (otherwise, Step 5a would succeed). Now, for all $x \in A - D_3$, $f(x) \neq \varphi_q(x)$. (Since that was the way D_3 was chosen in Step 2 of Stage s). Thus, $\varphi_q \neq^{err_{1,2}-1} f$.

From the above cases, it follows that $\mathcal{C} \notin \mathbf{Mul}_3^2\mathbf{N}^c\mathbf{Ex}^d$. □

The results established so far identified cases where multiple noisy texts are a restriction over learning from a single noisy text. The natural question that arises is what are the cases where the restriction of learning from multiple noisy texts can be compensated in some way. That is, for which values of a, b, c, d, $\mathbf{N}^a\mathbf{Ex}^b \subseteq \mathbf{Mul}_3^2\mathbf{N}^c\mathbf{Ex}^d$. A quick look at the effects of the various parameters yields at least two different ways in which the restrictive impact of multiple texts can be compensated. These are:

- by allowing extra errors in the program inferred from multiple texts (that is, by making d larger than b),
- by having a reduced bound on the number of noise elements in the multiple texts (that is, by making c smaller than a).

Theorems 5.9 and 5.10 use the first technique; Theorems 5.12, 5.13, and 5.14 use the second technique; and Theorem 5.11 can be seen as using a combination of the two techniques to compensate for the deleterious effects of learning from multiple texts. The technique used in these simulation arguments is to first collect all the input points where there is a noise in some text. Then construct a program which judiciously chooses the output on these inputs based on on the input texts and outputs of the converging programs.

Theorem 5.9 below shows that the collections of functions for which a b-error program can be identified from a single c-noisy text can also be learned from 3 texts, at least 2 of which are c-noisy, provided we are prepared to tolerate up to twice the number ($2b$) of errors in the final program.

Theorem 5.9. *Let $b, c \in N$. Then,* $\mathbf{N}^c\mathbf{Ex}^b \subseteq \mathbf{Mul}_3^2\mathbf{N}^c\mathbf{Ex}^{2b}$.

Proof. Suppose machine $\mathbf{M}$ $\mathbf{N}^c\mathbf{Ex}^b$-identifies $\mathcal{C}$. We construct $\mathbf{M}'$ such that $\mathbf{M}'$ $\mathbf{Mul}_3^2\mathbf{N}^c\mathbf{Ex}^{2b}$-identifies $\mathcal{C}$.

Let T_1, T_2, T_3 be the three given texts out of which at least two are c-noisy for $f \in \mathcal{C}$.

Let i, j be distinct and such that: $\mathbf{M}(T_i)\downarrow$ and $\mathbf{M}(T_j)\downarrow$, $P_i = \mathbf{M}(T_i)$ and $P_j = \mathbf{M}(T_j)$ and $\mathrm{card}(\{x \mid \varphi_{P_i}(x)\downarrow \neq \varphi_{P_j}(x)\downarrow\}) \leq 2b$. Let $CLASH = \{x \mid \varphi_{P_i}(x)\downarrow \neq \varphi_{P_j}(x)\downarrow\}$. Let $S \subseteq CLASH$ be a set of cardinality $\lfloor \frac{\mathrm{card}(CLASH)}{2} \rfloor$.

Note that we can easily determine all the sets defined above and P_i, P_j as above in the limit from the given texts.

$\mathbf{M}'$ in the limit outputs a program p such that:

$$\varphi_p(x) = \begin{cases} \varphi_{P_i}(x), & \text{if } x \in S; \\ \varphi_{P_j}(x), & \text{if } x \in CLASH - S; \\ \varphi_{P_i}(x), & x \in N - CLASH \ \wedge \ x \in \mathrm{domain}(\varphi_{P_i}); \\ \varphi_{P_j}(x), & x \in N - CLASH \ \wedge \ x \in \mathrm{domain}(\varphi_{P_j}); \\ \uparrow, & \text{otherwise.} \end{cases}$$

Since at least one of P_i, P_j is a b-error program for f, it is clear that p is a $2b$-error program for f. Thus, $\mathbf{M}'$ $\mathbf{Mul}_3^2\mathbf{N}^c\mathbf{Ex}^{2b}$-identifies f. □

In order to prove the rest of the results in this section (Theorems 5.10 to 5.14), it is useful to define certain sets and texts. We will require these sets to satisfy certain assumptions which can be assumed *without loss of generality*. We state these definitions and assumptions next, and we assume them without explicitly stating them in the proofs.

Suppose a machine is trying to $\mathbf{Mul}_3^2\mathbf{N}^c\mathbf{Ex}^d$-identify f (where $c \in N$). Suppose the three texts provided as input to the machine are T_1, T_2, and T_3, such that at least two of these texts are c-noisy for f.

Definitions: Let T_1, T_2, and T_3 be texts.

1. T_{UNION} is a text such that $\text{content}(T_{UNION}) = \{(x,y) \mid (x,y) \in \text{content}(T_1) \ \vee \ (x,y) \in \text{content}(T_2) \ \vee \ (x,y) \in \text{content}(T_3)\}$.
2. $NOISE\_POINTS = \{x \mid (\exists y,z)[y \neq z \ \wedge \ (x,y) \in \text{content}(T_{UNION}) \ \wedge \ (x,z) \in \text{content}(T_{UNION})]\}$.
3. $NOISE_{1,2} = \{x \mid \text{card}(\{y \mid (x,y) \in \text{content}(T_1)\}) = \text{card}(\{y \mid (x,y) \in \text{content}(T_2)\}) \geq 2 \ \wedge \ \text{card}(\{y \mid (x,y) \in \text{content}(T_3)\}) = 1\}$.
4. $NOISE_{1,3} = \{x \mid \text{card}(\{y \mid (x,y) \in \text{content}(T_1)\}) = \text{card}(\{y \mid (x,y) \in \text{content}(T_3)\}) \geq 2 \ \wedge \ \text{card}(\{y \mid (x,y) \in \text{content}(T_2)\}) = 1\}$.
5. $NOISE_{2,3} = \{x \mid \text{card}(\{y \mid (x,y) \in \text{content}(T_2)\}) = \text{card}(\{y \mid (x,y) \in \text{content}(T_3)\}) \geq 2 \ \wedge \ \text{card}(\{y \mid (x,y) \in \text{content}(T_1)\}) = 1\}$.
6. $NOISE_{1,2,3} = NOISE\_POINTS - (NOISE_{1,2} \cup NOISE_{1,3} \cup NOISE_{2,3})$.

Assumptions: We make the following assumptions about the notions defined above.

1. For all $i \in \{1,2,3\}$, $\text{card}(\{x \mid (\exists y,z)[y \neq z \ \wedge \ (x,y) \in \text{content}(T_i) \ \wedge \ (x,z) \in \text{content}(T_i)]\}) \leq c$ (since if the cardinality of noise points is greater than c, then we know which two texts are c-noisy texts for f).
2. For all $i,j \in \{1,2,3\}$ and for all $x \in NOISE\_POINTS$, $\text{card}(\{y \mid (x,y) \in \text{content}(T_i) \cap \text{content}(T_j)\}) \geq 1$ (this is so because otherwise at least one of T_i, T_j is not a c-noisy text for f, which implies that the remaining text is a c-noisy text for f).
3. For all x, y, $\text{card}(\{i \mid (x,y) \in \text{content}(T_i)\}) \neq 1$ (otherwise, surely $f(x) \neq y$ and we can thus drop (x,y) from consideration).

Note that if the above assumptions do not hold then we can convert the texts (in the limit) to T_1', T_2' and T_3' such that at least two of these texts are c-noisy for f and T_1', T_2', T_3' satisfy the assumptions given above.

The next theorem is a variation on Theorem 5.9 where we compensate for multiple texts by allowing extra errors in the final program. In this case, instead of allowing twice the number of errors in the final program, the upper bound on the number of errors is expressed in terms of the error bound for single text, b, and the noise level of the single text, c. We would like to note that the simulation carried out in Theorem 5.10 just below is optimal in the following sense. If we keep the bound on the number of inaccuracies in the texts same for the single text and for the acceptable texts in the multiple case, and $c \leq b$, then for all the collections of functions for which a b-error program can be identified from a single c-noisy text, a $b + \lceil \frac{c}{2} \rceil$-error program can also be identified from 3 texts at least 2 of which are only c-noisy. The optimality of the result is in the sense that this simulation fails if we reduce the bound on the number of errors allowed in the final program (of the multiple case) by even 1. This was the diagonalization result described in Corollary 5.5.

THEOREM 5.10. *Let $b, c \in N$. Then,* $\mathbf{N}^c\mathbf{Ex}^b \subseteq \mathbf{Mul}_3^2\mathbf{N}^c\mathbf{Ex}^{b+\lceil \frac{c}{2} \rceil}$.

Proof. Suppose machine $\mathbf{M}$ $\mathbf{N}^c\mathbf{Ex}^b$-identifies $\mathcal{C}$. We construct $\mathbf{M}'$ such that $\mathbf{M}'$ $\mathbf{Mul}_3^2\mathbf{N}^c\mathbf{Ex}^{b+\lceil \frac{c}{2} \rceil}$-identifies $\mathcal{C}$. Let T_1, T_2, T_3 be three given texts out of which at least two are c-noisy for f.

Let $m = \min(\{\mathrm{card}(NOISE_{1,2}), \mathrm{card}(NOISE_{1,3}), \mathrm{card}(NOISE_{2,3})\})$.
Let S be a set of cardinality $3m$ such that $\mathrm{card}(S \cap NOISE_{1,2}) = \mathrm{card}(S \cap NOISE_{1,3}) = \mathrm{card}(S \cap NOISE_{2,3})$.
Let $X \subseteq NOISE\_POINTS - S$ be a set of cardinality $\lceil \frac{\mathrm{card}(NOISE\_POINTS)-3m}{2} \rceil$.
Let i, j be distinct and such that: $\mathbf{M}(T_i)\downarrow$ and $\mathbf{M}(T_j)\downarrow$, $P_i = \mathbf{M}(T_i)$ and $P_j = \mathbf{M}(T_j)$ and $\mathrm{card}(\{x \mid \varphi_{P_i}(x)\downarrow \neq \varphi_{P_j}(x)\downarrow \ \wedge \ x \notin NOISE\_POINTS\}) \leq 2b$. Let $CLASH = \{x \mid \varphi_{P_i}(x)\downarrow \neq \varphi_{P_j}(x)\downarrow \ \wedge \ x \notin NOISE\_POINTS\}$.
Note that we can easily determine all the sets defined above and P_i, P_j as above in the limit from the given texts. $\mathbf{M}'$ in the limit outputs a program p such that:

$$\varphi_p(x) = \begin{cases} \varphi_{P_i}(x), & \text{if } x \in X; \\ \varphi_{P_j}(x), & \text{if } x \in NOISE\_POINTS - (X \cup S); \\ y, & \text{if } x \in CLASH \ \wedge \ (x,y) \in \mathrm{content}(T_{UNION}); \\ \varphi_{P_i}(x), & \text{if } x \in N - (NOISE\_POINTS \cup CLASH) \text{ and} \\ & x \in \mathrm{domain}(\varphi_{P_i}); \\ \varphi_{P_j}(x), & \text{if } x \in N - (NOISE\_POINTS \cup CLASH) \text{ and} \\ & x \in \mathrm{domain}(\varphi_{P_j}); \\ y, & \text{if } x \in S \ \wedge \ (x,y) \in \mathrm{content}(T_1) \cap \mathrm{content}(T_2) \cap \mathrm{content}(T_3); \\ \uparrow, & \text{otherwise.} \end{cases}$$

Since at least one of P_i, P_j computes a b-variant of f, the number of errors committed by the φ program $p \leq b + m + \lceil \frac{\mathrm{card}(NOISE\_POINTS-S)}{2} \rceil$, which is at most $b + \lceil \frac{c}{2} \rceil$. □

The next theorem uses a combination of increasing the bound on the number of errors in the final program and having a smaller noise level than in the case of single text to compensate for the restrictive effects of multiple texts. The thoerem shows that given $b, c, d \in N$ such that b is smaller, but not too smaller, than c (that is, $\frac{c}{2} \leq b \leq c$), all the collections of functions for which a b-error program can identified from a single $(2c-1)$-noisy text can also be identified from 3 texts, at least 2 of which are c-noisy, provided we are prepared to tolerate a bound of $\geq \max(\{\frac{4b}{3}, 2b - \frac{c}{2}\})$ on the number of errors in the final program. (Compare the following theorem with Corollary 5.6).

Theorem 5.11. *Let $b, c, d \in N, \frac{c}{2} \leq b \leq c$ be given. Then,* $\mathbf{N}^{2c-1}\mathbf{Ex}^b \subseteq \mathbf{Mul}^2_3\mathbf{N}^c\mathbf{Ex}^d$ *if $d \geq \max(\{\frac{4b}{3}, 2b - \frac{c}{2}\})$.*

Proof. Suppose machine $\mathbf{M}$ $\mathbf{N}^{2c-1}\mathbf{Ex}^b$-identifies $\mathcal{C}$. We construct machine $\mathbf{M}'$ such that $\mathbf{M}'$ $\mathbf{Mul}^2_3\mathbf{N}^c\mathbf{Ex}^d$-identifies $\mathcal{C}$. Let T_1, T_2, T_3 be three texts given for a function $f \in \mathcal{C}$, such that at least two of the three texts are c-noisy for f.

Note that T_{UNION} is a noisy text for f. If T_{UNION} is a $(2c-1)$-noisy text for f, then $\mathbf{M}'$ can just output the programs output by $\mathbf{M}$ on T_{UNION} to identify f with at most d errors. Thus, assume that T_{UNION} so formed contains more than $2c-1$ noisy elements. A simple calculation shows that the amount of noise in T_{UNION} is bounded by $\frac{3c+\mathrm{card}(NOISE_{1,2,3})}{2}$. If $\mathrm{card}(NOISE_{1,2,3}) < c$ then T_{UNION} is a $(2c-1)$-noisy text for f. Thus, $\mathrm{card}(NOISE_{1,2,3}) = c$. Also, for all $x \in NOISE_{1,2,3}$, we can assume that there exist distinct y_1^x, y_2^x, y_3^x such that $(x, y_1^x) \in \mathrm{content}(T_2) \cap \mathrm{content}(T_3)$, $(x, y_2^x) \in \mathrm{content}(T_1) \cap \mathrm{content}(T_2)$, and $(x, y_3^x) \in \mathrm{content}(T_1) \cap \mathrm{content}(T_3)$ (otherwise T_{UNION} contains at most $2c-1$ noisy elements).

Let i, j be distinct and such that: $\mathbf{M}(T_i)\downarrow$ and $\mathbf{M}(T_j)\downarrow$, $P_i = \mathbf{M}(T_i)$ and $P_j = \mathbf{M}(T_j)$ and $\mathrm{card}(\{x \mid \varphi_{P_i}(x)\downarrow \neq \varphi_{P_j}(x)\downarrow \ \wedge \ x \notin NOISE\_POINTS\}) \leq 2b$. Let $CLASH = \{x \mid [\varphi_{P_i}(x)\downarrow \neq \varphi_{P_j}(x)\downarrow] \ \wedge \ [x \notin NOISE\_POINTS]\}$. For $m \in \{i, j\}$, let

$\delta_m = \mathrm{card}(\{x \mid x \in \mathit{NOISE\_POINTS} \wedge [(x, \varphi_{P_m}(x)) \in \mathrm{content}(T_i) \cap \mathrm{content}(T_j)]\})$. Let $\delta = \min(\delta_i, \delta_j)$. We can suppose without loss of generality that $c - b \leq \delta \leq b$ (otherwise we can determine a text which is guaranteed to be c-noisy for f. For example, suppose that $\delta = \delta_i < c - b$, then clearly one of T_i and T_j is not a c-noisy text for f. This implies that the remaining text is a c-noisy text for f. Other cases are similar. In any of these cases, $\mathbf{M}'$ can simply input the correct c-noisy text obtained by the above analysis into $\mathbf{M}$ and arrive (in the limit) at a program which is at most b ($\leq d$) variant of f).

We will pick a $\mu \in N, \mu \leq c$ depending on δ. Let $S \subseteq \mathit{NOISE\_POINTS}$ be a set of cardinality μ. Let $X \subseteq \mathit{NOISE\_POINTS} - S$ be a set of cardinality $\lceil \frac{c-\mu}{2} \rceil$.

$\mathbf{M}'$ in the limit outputs a program p_μ such that the following holds:

$$\varphi_{p_\mu}(x) = \begin{cases} y_k^x, & \text{if } x \in S \ \wedge \ k \in \{1,2,3\} - \{i,j\}; \\ y_i^x, & \text{if } x \in X; \\ y_j^x, & \text{if } x \in \mathit{NOISE\_POINTS} - (X \cup S); \\ y, & \text{if } x \in \mathit{CLASH} \ \wedge \ (x,y) \in \mathrm{content}(T_{UNION}); \\ \varphi_{P_i}(x), & \text{if } x \in N - (\mathit{NOISE\_POINTS} \cup \mathit{CLASH}) \text{ and} \\ & x \in \mathrm{domain}(\varphi_{P_i}); \\ \varphi_{P_j}(x), & \text{if } x \in N - (\mathit{NOISE\_POINTS} \cup \mathit{CLASH}) \text{ and} \\ & x \in \mathrm{domain}(\varphi_{P_j}); \\ \uparrow, & \text{otherwise.} \end{cases}$$

Now, the number of errors committed by $\mathbf{M}'$ on f is bounded by $\mathit{maxerr} = \max(\{b - \mu + \delta, b - \delta + \mu + \lceil \frac{c-\mu}{2} \rceil\}) = \lceil \max(\{b - \mu + \delta, b - \delta + \mu + \frac{c-\mu}{2}\}) \rceil$. By choosing $\mu = \max(\{0, \lfloor \frac{4\delta - c}{3} \rfloor\})$ we have $\mathit{maxerr} = \lceil \max(\{b + \frac{c-\delta}{3}, b - \delta + \frac{c}{2}\}) \rceil$. The value of maxerr is maximized for $\delta = c - b$. Thus, $\mathit{maxerr} = \lceil \max(\{\frac{4b}{3}, 2b - \frac{c}{2}\}) \rceil$. Thus, if $d \geq \lceil \max(\{\frac{4b}{3}, 2b - \frac{c}{2}\}) \rceil$, then $\mathbf{N}^{2c-1}\mathbf{Ex}^b \subseteq \mathbf{Mul}_3^2\mathbf{N}^c\mathbf{Ex}^d$. □

We now present three theorems that use the technique of making the bound on the noise level of multiple texts smaller than the bound on the noise level of the single text to compensate for the restrictive effect of multiple texts.

The result, just below, says that for $b, c \in N$ such that b is fairly small compared to c (that is, $b < \lceil \frac{c}{2} \rceil$), if for some collection of functions a b-error program can be identified from a single $(2c-1)$-noisy text, then a b-error program can also be identified from 3 texts, at least 2 of which are c-noisy.

Theorem 5.12. *For* $b < \lceil \frac{c}{2} \rceil$, $\mathbf{N}^{2c-1}\mathbf{Ex}^b \subseteq \mathbf{Mul}_3^2\mathbf{N}^c\mathbf{Ex}^b$.

Proof. Suppose machine $\mathbf{M}$ $\mathbf{N}^{2c-1}\mathbf{Ex}^b$-identifies $\mathcal{C}$. We describe $\mathbf{M}'$ such that $\mathbf{M}'$ $\mathbf{Mul}_3^2\mathbf{N}^c\mathbf{Ex}^b$-identifies $\mathcal{C}$. Let T_1, T_2, T_3 be the three given texts out of which at least two are c-noisy for $f \in \mathcal{C}$.

As in Theorem 5.11, it can be shown that either T_{UNION}, is a $(2c-1)$-noisy for f or else there exists a set X of cardinality c and distinct y_1^x, y_2^x, y_3^x, for each $x \in X$, such that $(\forall x \in X)(\forall i \in \{1,2,3\})(\forall y \in \{y_1^x, y_2^x, y_3^x\} - \{y_i^x\})[(x,y) \in \mathrm{content}(T_i)]$. Now, let $i \in \{1,2,3\}$ be such that $\mathbf{M}(T_i)\downarrow$. Let $\mathbf{M}(T_i) = P_i$. For $j \in \{1,2,3\} - \{i\}$, let $D_j = \{x \in X \mid \varphi_{P_i}(x) = y_j^x\}$. Let $j \in \{1,2,3\} - \{i\}$ be such that $\mathrm{card}(D_j) \leq \frac{c}{2}$. Note that such a j exists. Now, for all $x \in X$, $f(x) \neq y_j^x$ (since otherwise T_i is a c-noisy text for f and thus φ_{P_i} should be a b-variant of f). Hence, T_j is a c-noisy text for f. Thus $\mathbf{M}(T_j)\downarrow$ and $\varphi_{\mathbf{M}(T_j)}$ is a b-variant of f. Clearly, in all the above cases $\mathbf{M}'$ can in the limit easily output a program which is a b-variant of f. □

The next theorem is like the previous one, but it does not require that error bound on the final program, b, be fairly small compared to the bound on the noise level of the multiple texts, c. It says that for $b, c \in N$, if for some collection of functions a b-error program can be identified from a single $(2c)$-noisy text, then a b-error program can also be identified from 3 texts, at least 2 of which are c-noisy.

THEOREM 5.13. *Let* $b, c \in N$. $\mathbf{N}^{2c}\mathbf{Ex}^{b} \subseteq \mathbf{Mul}_{3}^{2}\mathbf{N}^{c}\mathbf{Ex}^{b}$.

Proof. Suppose machine $\mathbf{M}$ $\mathbf{N}^{2c}\mathbf{Ex}^{b}$-identifies $\mathcal{C}$. We construct $\mathbf{M}'$ such that $\mathbf{M}'$ $\mathbf{Mul}_{3}^{2}\mathbf{N}^{c}\mathbf{Ex}^{b}$-identifies $\mathcal{C}$.

Let T_1, T_2, T_3 be three texts given for a function $f \in \mathcal{C}$, such that at least two of the three texts are c-noisy for f. A simple calculation shows that the amount of noise in T_{UNION} (with respect to the graph of f) is bounded by $\frac{3c+\mathrm{card}(NOISE_{1,2,3})}{2}$. Noting that $\mathrm{card}(NOISE_{1,2,3})$ is bounded by c, it follows that T_{UNION} is a $2c$-noisy text for f. $\mathbf{M}'$ outputs in the limit whatever is output by $\mathbf{M}$, in the limit, on the text T_{UNION}. Thus, $\mathbf{M}'$ $\mathbf{Mul}_{3}^{2}\mathbf{N}^{c}\mathbf{Ex}^{b}$-identifies f. This proves the theorem. □

Using a proof technique similar to the above, we can also establish the following variation of the above result. Here the bound on the noise level of the single text is expressed in terms of both the bound in the noise level of the multiple text (c) and the bound on the number of errors in the final program (b). It says that for $b, c \in N$, if for some collection of functions a b-error program can be identified from a single $\frac{3c+2b}{2}$-noisy text, then a b-error program can also be identified from 3 texts, at least 2 of which are c noisy. We omit the proof.

THEOREM 5.14. *Let* $b, c \in N$. *Then,* $\mathbf{N}^{\frac{3c+2b}{2}}\mathbf{Ex}^{b} \subseteq \mathbf{Mul}_{3}^{2}\mathbf{N}^{c}\mathbf{Ex}^{b}$.

5.2.2. Incomplete Texts. In this section we consider the effects of learning from multiple texts on incomplete data. We first show a result that gives cases in which identification from multiple incomplete texts is a restriction on identification from a single incomplete text. We then give results which show the cases where the restrictive effects of multiple inaccurate texts can be compensated. Unfortunately, our results about incomplete texts are not as extensive as those for noisy texts.

The following theorem shows that given b, c such that $b \geq c > 1$, there are collections of functions for which a b-error program can be identified from a single $(\lfloor \frac{3c}{2} \rfloor - 1)$-incomplete text, but for which even a $(b + \lceil \frac{c}{2} \rceil - 1)$-error program cannot be identified from 3 texts, at least 2 of which are only c-incomplete for the function being learned.

THEOREM 5.15. Let $b, c \in N, b \geq c > 1$. Then, $\mathbf{In}^{\lfloor \frac{3c}{2} \rfloor - 1}\mathbf{Ex}^{b} - \mathbf{Mul}_{3}^{2}\mathbf{In}^{c}\mathbf{Ex}^{b+\lceil \frac{c}{2} \rceil - 1} \neq \emptyset$.

Proof. The proof uses ideas similar to those used in the proof of Theorem 5.4. Suppose b, c are given as in the hypothesis. Let $A = \{\langle 0, x\rangle \mid 0 \leq x < \lfloor \frac{3c}{2} \rfloor\}$, $A_1 = \{\langle 0, x\rangle \mid 0 \leq x < \lfloor \frac{c}{2} \rfloor\}$, $A_2 = \{\langle 0, x\rangle \mid \lfloor \frac{c}{2} \rfloor \leq x < 2\lfloor \frac{c}{2} \rfloor\}$, $A_3 = \{\langle 0, x\rangle \mid 2\lfloor \frac{c}{2} \rfloor \leq x < 3\lfloor \frac{c}{2} \rfloor\}$, and $A_4 = \{\langle 0, x\rangle \mid 3\lfloor \frac{c}{2} \rfloor \leq x < \lfloor \frac{3c}{2} \rfloor\}$.

Note that $\mathrm{card}(A_1) = \mathrm{card}(A_2) = \mathrm{card}(A_3) = \lfloor \frac{c}{2} \rfloor$ and $\mathrm{card}(A_4) = 1$ if c is odd and 0 otherwise.

Consider the following class of functions

$\mathcal{C} = \{f \in \mathcal{R} \mid$ the following conditions hold:

(1)$(\forall i \in \{1,2,3\})(\forall x \in A_i)[f(x) = f(\langle 0, (i-1)\lfloor \frac{c}{2} \rfloor\rangle) \in \{0,1\}]$;

(2)$(\exists! i \in \{1,2,3\})[f(A_i) = 1]$;

(3)$(\forall i \in \{1,2,3\} \mid f(A_i) = 0)[(\max(\{f(\langle i, \langle x, 0\rangle\rangle) \mid x \in N\})$ exists$)$ $\wedge$ $(\varphi_{\max(\{f(\langle i,\langle x,0\rangle\rangle) \mid x \in N\})} =^{b} f)]$;

(4)$(\forall i \in \{1,2,3\})(\forall x, y, z)[f(\langle i, \langle x, y\rangle\rangle) = f(\langle i, \langle x, z\rangle\rangle)]$;

(5)$(\forall x \in A_4)[\varphi_{f(x)} =^{b} f]$ $\}$

Clearly, $\mathcal{C} \in \mathbf{In}^{\lfloor \frac{3c}{2} \rfloor - 1}\mathbf{Ex}^{b}$. We now show that $\mathcal{C} \notin \mathbf{Mul}_{3}^{2}\mathbf{In}^{c}\mathbf{Ex}^{b+\lceil \frac{c}{2} \rceil - 1}$.

Suppose, by way of contradiction that machine $\mathbf{M}$ $\mathbf{Mul}_3^2\mathbf{In}^c\mathbf{Ex}^{b+\lceil\frac{c}{2}\rceil-1}$-identifies $\mathcal{C}$. Then, by the *operator recursion theorem* [5], there exists a recursive 1-1 increasing p, $p(0) > 1$, defined below in stages.

We define $\varphi_{p(i)}$ in Stages $s \geq 3$.

Let x_3^s denote the least x such that for all y, $\varphi_{p(1)}(\langle 3, \langle x, y\rangle\rangle)$ is not defined before Stage s.

Let $\varphi_{p(1)}(x) = 0$ for $x \in A_1 \cup A_3$; $\varphi_{p(1)}(x) = 1$, for $x \in A_2$; $\varphi_{p(1)}(\langle 1, \langle 0, 0\rangle\rangle) = p(1)$; $\varphi_{p(1)}(\langle 2, \langle 0, 0\rangle\rangle) = p(2)$; $\varphi_{p(1)}(x) = p(1)$ for $x \in A_4$.

Let $\varphi_{p(2)}(x) = 0$, for $x \in A_2 \cup A_3$; $\varphi_{p(2)}(x) = 1$, for $x \in A_1$; $\varphi_{p(2)}(\langle 1, \langle 0, 0\rangle\rangle) = p(1)$; $\varphi_{p(2)}(\langle 2, \langle 0, 0\rangle\rangle) = p(2)$; $\varphi_{p(2)}(x) = p(2)$ for $x \in A_4$.

Let σ_1^3, σ_2^3 and σ_3^3 be initial segments such that

content(σ_1^3) $= \{(x, 0) \mid x \in A_1\} \cup \{(\langle 1, \langle 0, 0\rangle\rangle, p(1)), (\langle 2, \langle 0, 0\rangle\rangle, p(2))\}$.

content(σ_2^3) $= \{(x, 0) \mid x \in A_2\} \cup \{(\langle 1, \langle 0, 0\rangle\rangle, p(1)), (\langle 2, \langle 0, 0\rangle\rangle, p(2))\}$.

content(σ_3^3) $= \{(x, 0) \mid x \in A_3\} \cup \{(\langle 1, \langle 0, 0\rangle\rangle, p(1)), (\langle 2, \langle 0, 0\rangle\rangle, p(2))\}$.

Let $\varphi_{p(i)}^s$ denote the part of $\varphi_{p(i)}$ defined before Stage s. Go to Stage 3.

Begin Stage s

1. Let $q = \mathbf{M}(\sigma_1^s, \sigma_2^s, \sigma_3^s)$.

2. Let $D_1 = \{x \mid \Phi_q(x) \leq s \wedge ((x \in A_1 \wedge \varphi_q(x) = 0) \vee (x \in A_2 \wedge \varphi_q(x) = 1) \vee (x \in A_4 \wedge \varphi_q(x) = p(1)))\}$. Let $D_2 = \{x \mid \Phi_q(x) \leq s \wedge ((x \in A_1 \wedge \varphi_q(x) = 1) \vee (x \in A_2 \wedge \varphi_q(x) = 0) \vee (x \in A_4 \wedge \varphi_q(x) = p(2)))\}$. If card($D_1$) $\geq$ card(D_2), then let $r = 1$ and $r' = 2$; otherwise, let $r = 2$ and $r' = 1$. Let $X^s \subset \{\langle 4, x\rangle \mid x \geq s\}$ denote the (lexicographically least) set of b elements such that, for all $x \in X^s$, $\varphi_{p(1)}(x)$ is not defined before Stage s.

3. Let $\varphi_{p(s)}(x) = 0$, for $x \in A_3 \cup A_{r'}$; $\varphi_{p(s)}(x) = 1$ for $x \in A_r$; $\varphi_{p(s)}(x) = p(r')$ for $x \in A_4$; $\varphi_{p(s)}(\langle 3, \langle x_3^s, 0\rangle\rangle) = \varphi_{p(1)}(\langle 3, \langle x_3^s, 0\rangle\rangle) = \varphi_{p(2)}(\langle 3, \langle x_3^s, 0\rangle\rangle) = p(s)$. For $x \in$ domain$(\varphi_{p(1)}^s) - A$, let $\varphi_{p(s)}(x) = \varphi_{p(1)}(x)$.

4. Let σ_i' be an extension of σ_i^s such that content(σ_i') $=$ content(σ_i^s) $\cup$ $\{(\langle 3, \langle x_3^s, 0\rangle\rangle, p(s))\}$.

5. Dovetail Steps 5a, 5b, and 5c until (if ever) one of Steps 5a, 5b succeeds and at least one iteration of the loop at Step 5c is completed. If Step 5a succeeds, (before Step 5b (if ever) succeeds), then complete the then current iteration of the repeat loop in Step 5c and go to Step 6a. If Step 5b succeeds (before Step 5a (if ever) succeeds), then complete the then current iteration of the repeat loop in Step 5c and go to Step 6b.

5a. Search for $\sigma_i'' \supseteq \sigma_i'$, $i \in \{1, 2, 3\}$ such that $\mathbf{M}(\sigma_1^s, \sigma_2^s, \sigma_3^s) \neq \mathbf{M}(\sigma_1'', \sigma_2'', \sigma_3'')$ and for all $j, j' \in \{1, 2, 3\}$, for all $x, y, z \in N$ following holds:

- •content(σ_j'') $-$ content(σ_j') $=$ content($\sigma_{j'}''$) $-$ content($\sigma_{j'}'$).
- •If $(\langle j, \langle x, y\rangle\rangle, v) \in$ content(σ_1'') and $(\langle j, \langle x, z\rangle\rangle, w) \in$ content(σ_1''), then $v = w$.
- •If $(x, y) \in$ (content(σ_1'') $-$ content(σ_1')), then $[x \notin A]$ $\wedge$ $[y = 0$ OR $x \in X^s$ OR [For some $i \in \{1, 2, 3\}$, $u, v \in N$: $x = \langle i, \langle u, v\rangle\rangle$ and

$\varphi_{p(1)}(\langle i, \langle u, 0\rangle\rangle)$ has been defined till now and $\varphi_{p(1)}(\langle i, \langle u, 0\rangle\rangle) = y$]]. Intuitively, this substep attempts to look for a mind change.

5b. Search for $x \in (A_1 \cup A_2) - (D_1 \cup D_2)$ such that $\varphi_q(x)\downarrow$.

5c. Let *Addednew* $= \emptyset$.
(Note: We use *Addednew* in order to identify the set of elements for which we extended $\varphi_{p(1)}$ in this step. *Addednew* will be used (in Steps 6a and 6b) provided at least one of Steps 5a or 5b succeeds).

```
Repeat
    Let $x$ be the least point not in $X^s$ such that $\varphi_{p(1)}(x)$ is not defined till now. If, for some $i \in \{1, 2, 3\}$, $x$ is of the form $\langle i, \langle u, v\rangle\rangle$ and $\varphi_{p(1)}(\langle i, \langle u, 0\rangle\rangle)$ is defined till now, then let $\varphi_{p(1)}(x) = \varphi_{p(2)}(x) = \varphi_{p(s)}(x) = \varphi_{p(1)}(\langle i, \langle u, 0\rangle\rangle)$; else, let $\varphi_{p(1)}(x) = \varphi_{p(2)}(x) = \varphi_{p(s)}(x) = 0$.
    Let Addednew = Addednew $\cup \{x\}$.
Forever
```

6a. For all x, y such that $(x, y) \in \text{content}(\sigma_1'') - \text{content}(\sigma_1^s)$, let $\varphi_{p(1)}(x) = \varphi_{p(2)}(x) = y$. Let σ_i^{s+1} be an extension of σ_i'' such that $\text{content}(\sigma_i^{s+1}) - \text{content}(\sigma_i'') = \{(x, \varphi_{p(1)}(x)) \mid x \in \textit{Addednew}\}$. Go to Stage $s + 1$.

6b. Let σ_i^{s+1} be an extension of σ_i' such that $\text{content}(\sigma_i^{s+1}) - \text{content}(\sigma_i') = \{(x, \varphi_{p(1)}(x)) \mid x \in \textit{Addednew}\}$. Go to Stage $s + 1$.

End Stage s

Now consider the following cases:

Case 1: All stages terminate.

Let

$$f(x) = \begin{cases} 0, & \text{if } x \in A_1 \cup A_2; \\ 1, & \text{if } x \in A_3; \\ \varphi_{p(1)}(x), & \text{otherwise.} \end{cases}$$

Clearly, $f \in \mathcal{C}$ (since $c \leq b$, $\varphi_{p(1)} =^b f$, and $\varphi_{p(2)} =^b f$). Also, $\mathbf{M}$ on f makes infinitely many mind changes since Step 5b can succeed at most finitely many times before Step 5a succeeds.

Case 2: Stage s (≥ 3) is the least stage which never terminates.

In this case domain($\varphi_{p(s)}$) $= N - X^s$. Let q, D_1, D_2, r, r' be as defined in Steps 1, 2 of Stage s. Consider any total extension f of $\varphi_{p(s)}$ such that $(\forall x \in X^s)[\varphi_q(x) \neq f(x)]$.

Clearly, $f \in \mathcal{C}$ (since $\varphi_{p(r')} =^b f$ and $\varphi_{p(s)} =^b f$). Let T_1, T_2, T_3 be extensions of $\sigma_1', \sigma_2', \sigma_3'$ (as defined in Step 4 in Stage s) respectively such that $\text{content}(T_1) - \text{content}(\sigma_1^s) = \{(x, y) \mid f(x) = y \ \wedge \ (\forall z)[(x, z) \notin \text{content}(\sigma_1^s)]\}$ and, for all j, $[|\sigma_1'| + j]^{th}$ element of T_1, $[|\sigma_2'| + j]^{th}$ element of T_2 and $[|\sigma_3'| + j]^{th}$ element of T_3 are all same. First, notice that, for all such f and corresponding T_1, T_2, T_3: $\mathbf{M}(T_1, T_2, T_3) = q$ (otherwise Step 5a would succeed). Second, notice that, for all $x \in (A_1 \cup A_2) - (D_1 \cup D_2)$, $\varphi_q(x)\uparrow$ (otherwise Step 5b would succeed). Third, from the choice of r, r', it follows that program q makes at least $\lceil \frac{c}{2} \rceil$ errors on $A_1 \cup A_2 \cup A_4$. Also, from the way f has been chosen, we have $(\forall x \in X^s)[f(x) \neq \varphi_q(x)]$. Thus, $\varphi_q \neq^{b + \lceil \frac{c}{2} \rceil - 1} f$.

From the above cases, it follows that $\mathcal{C} \notin \mathbf{Mul}_3^2\mathbf{In}^c\mathbf{Ex}^{b+\lceil \frac{c}{2} \rceil -1}$. □

The above theorem demonstrated cases in which identification from multiple inaccurate texts is a restriction over identification from a single inaccurate text. Like in the case with noisy texts, we would like to explore situations in which the deleterious effects of learning from multiple texts can be compensated. Again, like in the case of noisy texts, this can be achieved in two different ways:

- by increasing the bound on the number of errors in the final program allowed in identification from multiple texts;
- by increasing the bound on missing data in the single incomplete text (this has the same effect as decreasing the bound on missing data in the multiple texts).

Theorem 5.16 just below uses the second technique and Theorems 5.19 and 5.20 use the first technique to compensate for the effects of multiple texts. The technique used in these simulation arguments is to first collect the input points on which the converging programs differ; then to construct a program which judiciously chooses the output on these inputs based on the input texts and outputs of the converging programs.

THEOREM 5.16. $\mathbf{In}^{c+\lfloor \frac{c}{2} \rfloor}\mathbf{Ex}^b \subseteq \mathbf{Mul}_3^2\mathbf{In}^c\mathbf{Ex}^b$.

The above theorem says that given b, c, if for a collection of functions a b-error program can be identified from a single $(c + \lfloor \frac{c}{2} \rfloor)$-incomplete text, then a b-error program can also be identified from 3 texts, at least 2 of which are c-incomplete for the function being learned. A proof of this result requires the notion of stabilizing sequence which we introduce next. Let L range over sets of ordered pairs.

DEFINITION 5.17. [9] σ is said to be a stabilizing sequence for $\mathbf{M}$ on $L \iff$ $[\text{content}(\sigma) \subseteq L$ and $(\forall \tau \mid \sigma \subset \tau \ \wedge \ \text{content}(\tau) \subseteq L)[\mathbf{M}(\tau) = \mathbf{M}(\sigma)]]$.

The following lemma is based on a similar lemma by Blum and Blum [3] (see also Osherson and Weinstein [18]).

LEMMA 5.18. (Based on a similar lemma in [3, 18]) *Suppose* $\mathbf{M}$ $\mathbf{In}^a\mathbf{Ex}^b$*-identifies* f*. Let* L *be such that* $L \subseteq \{(x, y) \mid f(x) = y\} \ \wedge \ \text{card}(\{(x, y) \mid f(x) = y\} - L) \leq a$*. Then there exists a stabilizing sequence for* $\mathbf{M}$ *on* L*. Moreover, if* σ *is a stabilizing sequence for* $\mathbf{M}$ *on* L*, then* $\varphi_{\mathbf{M}(\sigma)} =^b f$*.*

We now give a proof of Theorem 5.16.

Proof. (Theorem 5.16) Suppose machine $\mathbf{M}$ $\mathbf{In}^{c+\lfloor \frac{c}{2} \rfloor}\mathbf{Ex}^b$-identifies $\mathcal{C}$. We construct $\mathbf{M}'$ such that $\mathbf{M}'$ $\mathbf{Mul}_3^2\mathbf{In}^c\mathbf{Ex}^b$-identifies $\mathcal{C}$.

Let T_1, T_2, T_3 be three given texts out of which at least two are c-incomplete for $f \in \mathcal{C}$. Let T_{UNION} be a text such that $\text{content}(T_{UNION}) = \text{content}(T_1) \cup \text{content}(T_2) \cup \text{content}(T_3)$. Without loss of generality we assume that $(\forall x)[\text{card}(\{y \mid (x, y) \in \text{content}(T_{UNION})\}) \leq 1]$ (for if there are such x, y, z, i, j such that $(x, y) \in \text{content}(T_i), (x, z) \in \text{content}(T_j)$ and $y \neq z$, then at least one of T_i, T_j is not a c-incomplete text for f. Thus, the third text is a c-incomplete text for f). We can also assume without loss of generality that $(\forall x)[\text{card}(i \in \{1, 2, 3\} \mid (\exists y)[(x, y) \in \text{content}(T_i)]) \neq 2]$ (otherwise for such x's the value of $f(x)$ is known and thus without loss of generality we can put (x, y) in the content of all three texts).

In the following, we try to locate a stabilizing sequence for $\mathbf{M}$ on an appropriate subset of $\{(x, f(x)) \mid x \in N\}$. If one of the texts has a 'high amount of incompleteness,' then this may not be possible (see condition (iii) below); however, in this case it is possible to determine the text which has a high amount of incompleteness.

Let $n', n'' \in N$, $X \subset \{0, 1, \ldots, n'\}$ and $\sigma \in \text{SEQ}$ be such that, (i), (ii) and (iii) below are satisfied.

(i) $(\forall i \in \{1,2,3\})(\forall y \in N)(\forall x \in \{0,1,\ldots,n'\})[(x,y) \in \text{content}(T_i) \Rightarrow (x,y) \in \text{content}(T_i[n''])]$.
(ii) $(\forall x \in X)(\forall y \in N)[\text{card}(\{i \mid (x,y) \in \text{content}(T_i)\}) \leq 1]$.
(iii)
(A) $[(\exists j \in \{1,2,3\})[\text{card}(\{x \mid (x \in X) \ \wedge \ (\forall y \in N)[(x,y) \notin \text{content}(T_j)]\}) > c]]$
OR
(B) $[[[\text{content}(\sigma) \subseteq [\text{content}(T_{UNION}) \cap \{(x,y) \mid (y \in N) \ \wedge \ (x \leq n')\}] - [\{(x,y) \mid (x \in X) \ \wedge \ (y \in N)\}]]$ and σ is a stabilizing sequence for $\mathbf{M}$ on $L = [\text{content}(T_{UNION}) - \{(x,y) \mid (x \in X) \ \wedge \ (y \in N)\}]]$ AND $[(\forall x \leq n')[x \notin X \Rightarrow (\exists y)[\text{card}(\{i \mid (x,y) \in \text{content}(T_i)\}) = 3]]]]$.

It is easy to see that if such n', n'', X, σ exist then $\mathbf{M}'$ can find them in the limit. If in (iii) (A) holds then clearly $\mathbf{M}'$ can know the two texts which are c-incomplete for f, and thus output in the limit a b error program for f. If in (iii) (B) holds then we claim that $\mathbf{M}(\sigma)$ is a b-error program for f. This would be so if $\text{card}(\{(x,y) \mid f(x) = y\} - L) \leq c + \lfloor \frac{c}{2} \rfloor$. To prove this, let $\delta_i = \text{card}(\{x \in X \mid (\exists y)[(x,y) \in \text{content}(T_i)]\})$. Let δ be the median of $\delta_1, \delta_2, \delta_3$. Now, $\text{card}(\{(x,y) \mid f(x) = y\} - L) \leq c - (\text{card}(X) - \delta) + \text{card}(X) \leq c + \delta$ and $\delta \leq \frac{c}{2}$.

To complete the proof we have to show that such n', n'', X, σ indeed exist.

Case 1: There exists an $i \in \{1,2,3\}$ such that $\text{card}(\{x \mid (\forall y)[(x,y) \notin \text{content}(T_i)]\}) > c$.

Let n' be such that there exists an i, $\text{card}(\{x \leq n' \mid (\forall y)[(x,y) \notin \text{content}(T_i)]\}) = c+1$. Let n'' be such that (i) above is satisfied. Let $X = \{x \leq n' \mid \text{card}(\{i \mid (\exists y)[(x,y) \in \text{content}(T_i)]\}) \leq 1\}$. Then it is easy to see that (i), (ii) and (iii) above are satisfied.

Case 2: There does not exist an $i \in \{1,2,3\}$ such that $\text{card}(\{x \mid (\forall y)[(x,y) \notin \text{content}(T_i)]\}) > c$.

Let $X = \{x \mid \text{card}(\{i \mid (\exists y)[(x,y) \in \text{content}(T_i)]\}) \leq 1\}$. Now, let $L = \{(x,y) \mid f(x) = y \ \wedge \ x \notin X\}$, let $\delta_i = \text{card}(\{x \in X \mid (\exists y)[(x,y) \in \text{content}(T_i)]\})$. Let δ be the median of $\delta_1, \delta_2, \delta_3$. Now, $\text{card}(\{(x,y) \mid f(x) = y\} - L) \leq c - (\text{card}(X) - \delta) + \text{card}(X) \leq c + \delta$ and $\delta \leq \frac{c}{2}$. Thus, there exists a stabilizing sequence σ for $\mathbf{M}$ on L. Let $n' = 1 + \max(X \cup \{x \mid (\exists y)[(x,y) \in \text{content}(\sigma)]\})$. Let n'' be chosen so as to satisfy (i) above. Clearly, n', n'', X and σ satisfy (i), (ii) and (iii).

The above cases prove the existence of σ, X, n', n'' satisfying (i),(ii) and (iii). □

We now present two results that compensate for the restrictive effects of multiple incomplete texts by allowing a larger bound on the number of errors allowed in the final program.

Theorem 5.19. *Let $b, c \in N$. Then,* $\mathbf{In}^c\mathbf{Ex}^b \subseteq \mathbf{Mul}^2_3\mathbf{In}^c\mathbf{Ex}^{2b}$. The above result, which is a counterpart of Theorem 5.9, says that the collections of functions for which a b-error program can be identified from a c-incomplete text can also be identified from 3 texts, at least 2 of which are c-incomplete, provided we are prepared to tolerate twice, $2b$, the number of errors in the final program. This theorem can be proved using a technique similar to the one used in the proof of Theorem 5.9. We omit the details.

The next result gives a different bound on the number of errors allowed in the final program.

Theorem 5.20. Let $b, c \in N, b \geq c$ be given. Then, $\mathbf{In}^c\mathbf{Ex}^b \subseteq \mathbf{Mul}^2_3\mathbf{In}^c\mathbf{Ex}^{b+\lceil \frac{c}{2} \rceil}$.

Proof. Suppose hypothesis. Suppose machine $\mathbf{M}$ $\mathbf{In}^c\mathbf{Ex}^b$-identifies $\mathcal{C}$. We construct $\mathbf{M}'$ such that $\mathbf{M}'$ $\mathbf{Mul}_3^2\mathbf{In}^c\mathbf{Ex}^{b+\lceil\frac{c}{2}\rceil}$-identifies $\mathcal{C}$.

Let T_1, T_2, T_3 be three texts given for a function $f \in \mathcal{C}$, such that at least two of the three texts are c-incomplete for f. Without loss of generality we can assume that $(\forall x)[\text{card}(\{y \mid (x,y) \in \text{content}(T_1) \cup \text{content}(T_2) \cup \text{content}(T_3)\}) \leq 1]$ (for if there is such an x, then let y, z, i, j be such that $(x,y) \in \text{content}(T_i), (x,z) \in \text{content}(T_j)$ and $y \neq z$, then at least one of T_i, T_j is not a c-incomplete text for f. Thus, the third text is a c-incomplete text for f).

Let T_{UNION} be a text such that $\text{content}(T_{UNION}) = \{(x,y) \mid (x,y) \in \text{content}(T_1) \ \vee \ (x,y) \in \text{content}(T_2) \ \vee \ (x,y) \in \text{content}(T_3)\}$. Let i, j be distinct and such that: $\mathbf{M}(T_i)\downarrow$ and $\mathbf{M}(T_j)\downarrow$, $P_i = \mathbf{M}(T_i)$ and $P_j = \mathbf{M}(T_j)$, and $\text{card}(CLASH = \{x \mid [\varphi_{P_i}(x)\downarrow \neq \varphi_{P_j}(x)\downarrow] \ \vee \ (\exists y)[[(x,y) \in \text{content}(T_{UNION})] \ \wedge \ [[\varphi_{P_i}(x)\downarrow \neq y] \ \vee \ [\varphi_{P_j}(x)\downarrow \neq y]]]\}) \leq 3b$.

Let $VIS\_IN = CLASH \cap \{x \mid \text{card}(\{i \mid (\exists y)(x,y) \in \text{content}(T_i)\}) \leq 1\}$. For $i \in \{1,2,3\}$, let $IN_i = VIS\_IN \cap \{x \mid (\exists y)[(x,y) \in \text{content}(T_i)]\}$. Let $m = \min(\text{card}(IN_1), \text{card}(IN_2), \text{card}(IN_3))$. Let $S \subseteq VIS\_IN$ be a set of cardinality $3m$ such that $\text{card}(S \cap IN_1) = \text{card}(S \cap IN_2) = \text{card}(S \cap IN_3)$. Let $X \subseteq VIS\_IN - S$ be a set of cardinality $\lceil\frac{\text{card}(VIS\_IN-S)}{2}\rceil$.

Machine $\mathbf{M}'$ outputs (in the limit) program p as follows :

$$\varphi_p(x) = \begin{cases} \varphi_{P_i}(x), & \text{if } x \notin CLASH \ \wedge \ x \in \text{domain}(\varphi_{P_i}); \\ \varphi_{P_j}(x), & \text{if } x \notin CLASH \ \wedge \ x \in \text{domain}(\varphi_{P_j}); \\ y, & \text{if } x \in ((CLASH - VIS\_IN) \cup S) \text{ and} \\ & (x,y) \in \text{content}(T_{UNION}) \ ; \\ \varphi_{P_i}(x), & \text{if } x \in X; \\ \varphi_{P_j}(x), & \text{if } x \in VIS\_IN - (S \cup X); \\ \uparrow, & \text{otherwise.} \end{cases}$$

Now the number of errors committed by $\mathbf{M}'$ on f is bounded by $m + \lceil\frac{\text{card}(VIS\_IN-S)}{2}\rceil + b$. Observing that $c \geq \text{card}(VIS\_IN) - m$, it follows that $\mathbf{M}'$ $\mathbf{Mul}_3^2\mathbf{In}^c\mathbf{Ex}^{b+\lceil\frac{c}{2}\rceil}$-identifies $\mathcal{C}$. □

We end this section by presenting a somewhat surprising result which says that for $a \in N$ the collections of functions for which a 1-error program can be identified from a single a-incomplete text is exactly the same as the collections of functions for which a 1-error program can be identified from 3 texts, at least 2 of which are a-incomplete for the function being learned.

THEOREM 5.21. *Let $a \in N$. Then,* $\mathbf{In}^a\mathbf{Ex}^1 = \mathbf{Mul}_3^2\mathbf{In}^a\mathbf{Ex}^1$.

Proof. $a = 1$ case follows from Theorem 5.16. Suppose $a > 1$. Suppose $\mathbf{M}$ $\mathbf{In}^a\mathbf{Ex}^1$-identifies $\mathcal{C}$. We give an $\mathbf{M}'$ such that $\mathbf{M}'$ $\mathbf{Mul}_3^2\mathbf{In}^a\mathbf{Ex}^1$-identifies $\mathcal{C}$. Let T_1, T_2, and T_3 be the three given texts out of which at least two are a-incomplete for $f \in \mathcal{C}$. Let T_{UNION} denote a text such that $\text{content}(T_{UNION}) = \text{content}(T_1) \cup \text{content}(T_2) \cup \text{content}(T_3)$. Without loss of generality assume that, for all x, $\text{card}(\{y \mid (x,y) \in \text{content}(T_{UNION})\}) \leq 1$ (otherwise let y, z, $y \neq z$ be such that $(x,y) \in \text{content}(T_i)$ and $(x,z) \in \text{content}(T_j)$ (for $i, j \in \{1,2,3\}$). Clearly, at least one of T_i, T_j is not an a-incomplete text for f. Thus, the remaining text is an a-incomplete text for f; in this case, $\mathbf{M}'$ can simply input the a-incomplete text for f to $\mathbf{M}$ to obtain a program (in the limit) which is 1-variant of f). Also, assume that $(\forall x, y)[\text{card}(\{i \in \{1,2,3\} \mid (x,y) \in \text{content}(T_i)\}) \neq 2]$ (since otherwise $f(x) = y$ and we can assume that (x, y) is in all the texts). Let i, j be distinct and such that $\mathbf{M}(T_i)\downarrow$ and $\mathbf{M}(T_j)\downarrow$ (clearly, for $f \in \mathcal{C}$ such i, j exists and $\mathbf{M}'$ can determine i, j in the limit). Without loss of generality let $i = 1$ and $j = 2$ (otherwise simply rename the texts). Let $\mathbf{M}(T_1) = P_1$ and $\mathbf{M}(T_2) = P_2$.

Let $CLASH12 = \{x \mid [\varphi_{P_1}(x)\downarrow \neq \varphi_{P_2}(x)\downarrow] \ \vee \ [\varphi_{P_1}(x)\downarrow \ \wedge \ (\exists y)[(x, y) \in \text{content}(T_2) \ \wedge \ y \neq \varphi_{P_1}(x)]] \ \vee \ [\varphi_{P_2}(x)\downarrow \ \wedge \ (\exists y)[(x, y) \in \text{content}(T_1) \ \wedge \ y \neq \varphi_{P_2}(x)]]\}$. Let $CLASH1T3 = \{x \mid [\varphi_{P_1}(x)\downarrow \ \wedge \ (\exists y)[(x, y) \in \text{content}(T_3) \ \wedge \ y \neq \varphi_{P_1}(x)]]\}$. Let $CLASH2T3 = \{x \mid [\varphi_{P_2}(x)\downarrow \ \wedge \ (\exists y)[(x, y) \in \text{content}(T_3) \ \wedge \ y \neq \varphi_{P_2}(x)]]\}$.

Now, consider the following cases (note that $\mathbf{M}'$ can also determine in the limit which of the following cases holds). In each of the following cases, we either give a correct text for f or a program for an 1-variant of f. For each of the case numbers $i > 1$, we assume without stating that all cases $< i$ do not hold.

Case 1: $\text{card}(CLASH12) > 2$.

In this case T_3 is an a-incomplete text for f.

Case 2: $\text{card}(CLASH1T3) > 1$.

In this case T_2 is an a-incomplete text for f.

Case 3: $\text{card}(CLASH2T3) > 1$.

In this case T_1 is an a-incomplete text for f.

Case 4: $\text{card}(CLASH12) = 0$.

In this case, let p be such that $\varphi_p = \varphi_{P_1} \cup \varphi_{P_2}$. Then, φ_p is a program for an 1-variant of f.

For cases 5 to 7 assume without loss of generality that, for all $x \in CLASH12 \cup CLASH1T3 \cup CLASH2T3$, $\varphi_{P_1}(x)\downarrow$ and $\varphi_{P_2}(x)\downarrow$. (since $CLASH12 \cup CLASH1T3 \cup CLASH2T3$ is finite, we can check in the limit if $\varphi_{P_1}(x), \varphi_{P_2}(x)$ is defined or not. If it is undefined then we can assume that it converges to an arbitrary value). Also, for all $x \in CLASH12 \cup CLASH1T3 \cup CLASH2T3$ we can assume that $(\forall y)[[(x, y) \in \text{content}(T_1) \Rightarrow y = \varphi_{P_1}(x)] \ \wedge \ [(x, y) \in \text{content}(T_2) \Rightarrow y = \varphi_{P_2}(x)]]$ (otherwise we can assume that y is the value the program converges to). Note that this assumption may cause the values of $CLASH12, CLASH1T3$ and $CLASH2T3$, as defined above, to change. It may also cause one of cases 1,2, 3 or 4 to hold; however, in this case the above analysis also holds.

Case 5: $\text{card}(CLASH1T3) = \text{card}(CLASH2T3) = 1$.

Then, for all $x \in N - CLASH12 \cup CLASH1T3 \cup CLASH2T3$ $(\exists i \in \{1, 2\})[\varphi_{P_i}(x)\downarrow = f(x)]$. A simple calculation shows that $\text{card}(CLASH12 \cup CLASH1T3 \cup CLASH2T3) \leq 3$. If $a \geq \text{card}(CLASH12 \cup CLASH1T3 \cup CLASH2T3)$, then text T such that $\text{content}(T) = \{(x, y) \mid x \in N - (CLASH12 \cup CLASH1T3 \cup CLASH2T3) \ \wedge \ (\exists i \in \{1, 2, 3\})[\varphi_{P_i}(x) = y]\}$ is an a-incomplete text for f. Otherwise $a = 2$ and $\text{card}(CLASH12 \cup CLASH1T3 \cup CLASH2T3) = 3$. In this case, if there exists $i \in \{1, 2, 3\}$ such that $(\forall x \in CLASH12 \cup CLASH1T3 \cup CLASH2T3)(\forall y)[(x, y) \notin \text{content}(T_i)]$, then each of the remaining texts are a-incomplete for f. Otherwise let p be a program such that

$$\varphi_p(x) = \begin{cases} U(x), & \text{if } x \in N - (CLASH12 \ \cup \ CLASH13 \ \cup \ CLASH23); \\ y, & \text{if } x \in CLASH12 \ \cup \ CLASH13 \ \cup \ CLASH23 \\ & \ \wedge \ (x, y) \in \text{content}(T_{UNION}). \end{cases}$$

where $U(x)$ denotes $(\varphi_{P_1} \cup \varphi_{P_2})(x)$. Then, p is program for an 1-variant of f.

Case 6: $\text{card}(CLASH1T3) = 1$ and $\text{card}(CLASH2T3) = 0$ (case of $\text{card}(CLASH2T3) = 1$ and $\text{card}(CLASH1T3) = 0$ is similar).

Let p be a program such that:

$$\varphi_p(x) = \begin{cases} U(x), & \text{if } x \in N - CLASH12; \\ \varphi_{P_2}(x), & \text{otherwise.} \end{cases}$$

where $U(x)$ denotes $(\varphi_{P_1} \cup \varphi_{P_2})(x)$. Then, p is program for an 1-variant of f. Since if T_2 is an a-incomplete text for f, then this is so. Otherwise number of errors committed by p is $\leq$ card$(CLASH12 - CLASH1T3) \leq 1$.

Case 7: card$(CLASH1T3) =$ card$(CLASH2T3) = 0$.

In this case, for all $(x, y) \in$ content(T_3), $f(x) = y$. This is so since, if T_3 is the correct text then it is certainly so; otherwise, for all $x \in N - CLASH12$, $\varphi_{P_1}(x) = f(x) \ \vee \ \varphi_{P_2}(x) = f(x)$. Thus, if $(x, y) \in$ content(T_3), then $f(x) = y$. Thus, $[\mathbf{M}(T_3)\downarrow]$ or [card$(\{x \mid (\forall y)(x, y) \notin \text{content}(T_3)\}) > a$]. Note that we can determine in the limit one of the clauses above which holds.

Case 7a: card$(\{x \mid (\forall y)(x, y) \notin \text{content}(T_3)\}) > a$.

In this case, both T_1 and T_2 are a-incomplete texts for f.

Case 7b: $\mathbf{M}(T_3)\downarrow = P_3$.

Let $CLASH13 = \{x \mid [\varphi_{P_1}(x)\downarrow \neq \varphi_{P_3}(x)\downarrow]\}$. Let $CLASH23 = \{x \mid [\varphi_{P_2}(x)\downarrow \neq \varphi_{P_3}(x)\downarrow]\}$.

Case 7b.1: card$(CLASH13) > 2$.

In this case T_2 is an a-incomplete text for f.

Case 7b.2: card$(CLASH23) > 2$.

In this case T_1 is an a-incomplete text for f.

For case 7b.3 to 7b.5, assume without loss of generality that, for all $x \in CLASH12 \cup CLASH13 \cup CLASH23$, $\varphi_{P1}(x)\downarrow, \varphi_{P_2}(x)\downarrow$ and $\varphi_{P_3}(x)\downarrow$. (since $CLASH12 \ \cup \ CLASH13 \ \cup \ CLASH23$ is finite, we can check in the limit if $\varphi_{P_1}(x), \varphi_{P_2}(x), \varphi_{P_3}(x)$ is defined or not. If it is not defined, then we can assume that it converges to an arbitrary value). Also, for all $x \in CLASH12 \ \cup \ CLASH13 \ \cup \ CLASH23$ we can assume that $(\forall y)[[(x, y) \in \text{content}(T_1) \Rightarrow y = \varphi_{P_1}(x)] \ \wedge \ [(x, y) \in \text{content}(T_2) \Rightarrow y = \varphi_{P_2}(x)]]$ (otherwise we can assume that y is the value the program converges to). Note that this assumption may cause the values of $CLASH12, CLASH13, CLASH23, CLASH1T3$ and $CLASH2T3$, as defined above, to change. It may also cause one of cases 1,2, 3, 4 , 5, 6, 7b.1 or 7b.2 to hold; however, in this case the above analysis also holds.

Case 7b.3: card$(CLASH13) = 0$.

In this case, let p be such that $\varphi_p = \varphi_{P_1} \cup \varphi_{P_3}$. Then, φ_p is a program for an 1-variant of f.

Case 7b.4: card$(CLASH23) = 0$.

In this case, let p be such that $\varphi_p = \varphi_{P_2} \cup \varphi_{P_3}$. Then, φ_p is a program for an 1-variant of f.

Case 7b.5: $0 <$ card$(CLASH13) < 3$ and $0 <$ card$(CLASH23) < 3$.

Then, $(\forall x \in N - CLASH12 \cup CLASH13 \cup CLASH23)$ $(\exists i \in \{1, 2, 3\})[\varphi_{P_i}(x)\downarrow = f(x)]$. A simple calculation shows that card$(CLASH12 \ \cup \ CLASH13 \ \cup \ CLASH23) \leq 3$. If $a \geq$ card$(CLASH12 \ \cup \ CLASH13 \ \cup \ CLASH23)$, then text T such that content$(T) = \{(x, y) \mid x \in N - (CLASH12 \ \cup \ CLASH13 \ \cup \ CLASH23) \wedge (\exists i \in \{1, 2, 3\})[\varphi_{P_i}(x) = y]\}$ is an a-incomplete text for f. Otherwise $a = 2$ and card$(CLASH12 \cup CLASH13 \cup CLASH23) = 3$.

In this case, if there exists $i \in \{1,2,3\}$ such that $(\forall x \in CLASH12 \cup CLASH13 \cup CLASH23)(\forall y)[(x,y) \notin \text{content}(T_i)]$, then each of the remaining texts are a-incomplete texts for f. Otherwise let p be a program such that

$$\varphi_p(x) = \begin{cases} U(x), & \text{if } x \in N - (CLASH12 \cup CLASH13 \cup CLASH23); \\ y, & \text{if } x \in CLASH12 \ \cup \ CLASH13 \ \cup \ CLASH23 \\ & \wedge \ (x,y) \in \text{content}(T_{UNION}). \end{cases}$$

where $U(x)$ denotes $(\varphi_{P_1} \cup \varphi_{P_2} \cup \varphi_{P_3})(x)$. Then, p is program for an 1-variant of f. □

5.2.3. Imperfect Texts. The study of imperfect texts turns out to be very complex. We only present a simulation result which says that for $b, c \in N$, the collections of functions for which a b-error program can be identified from a single c-imperfect text, can also be learned from 3 texts, at least 2 of which are c-incomplete, provided we are prepared to tolerate up to twice the number of errors in the final program. A proof of this result can be worked out on similar lines to our proof of Theorem 5.9; we omit the details.

Theorem 5.22. *Let $b, c \in N$.* $\mathbf{Im}^c\mathbf{Ex}^b \subseteq \mathbf{Mul}^2_3\mathbf{Im}^c\mathbf{Ex}^{2b}$.

5.3. Hierarchy Results. We now turn our attention to hierarchy results for each of the inaccuracy types. In particular we show, for each inaccuracy type, larger collections of functions become identifiable if we

- increase the bound on the number of errors allowed in the final program (keeping other parameters the same), or
- decrease the bound on the number of inaccuracies allowed in the input texts (keeping other parameters the same).

The following two theorems, that follow from results about identification from single inaccurate texts [11], yield as corollaries the above hierarchies.

Theorem 5.23. *Let $b, j, k \in N$, $k \geq j > \lfloor \frac{k}{2} \rfloor$.*

(a) $\mathbf{Mul}^j_k\mathbf{Im}^*\mathbf{Ex}^{b+1} - \mathbf{Ex}^b \neq \emptyset$.

(b) $\mathbf{Mul}^j_k\mathbf{Im}^*\mathbf{Ex}^* - \bigcup_{b \in N} \mathbf{Ex}^b \neq \emptyset$.

Proof. Follows from Theorem 1 in [11] and Theorem 5.1. □

Theorem 5.24. $(\forall b, j, k \in N \mid k \geq j > \lfloor \frac{k}{2} \rfloor)$ $[\mathbf{Mul}^j_k\mathbf{Im}^b\mathbf{Ex} - [\mathbf{N}^{b+1}\mathbf{Ex}^* \cup \mathbf{In}^{b+1}\mathbf{Ex}^*] \neq \emptyset]$.

Proof. Follows from Theorem 6 in [11] and Theorem 5.2. □

It can be shown that the above two results yield the following two corollaries.

Corollary 5.25. *Let $a \in N \cup \{*\}$. Let $j, k \in N$ such that $k \geq j > \lfloor \frac{k}{2} \rfloor$. Then,*

1. $\mathbf{Mul}^j_k\mathbf{N}^a\mathbf{Ex}^0 \subset \mathbf{Mul}^j_k\mathbf{N}^a\mathbf{Ex}^1 \subset \cdots \subset \mathbf{Mul}^j_k\mathbf{N}^a\mathbf{Ex}^*$.
2. $\mathbf{Mul}^j_k\mathbf{In}^a\mathbf{Ex}^0 \subset \mathbf{Mul}^j_k\mathbf{In}^a\mathbf{Ex}^1 \subset \cdots \subset \mathbf{Mul}^j_k\mathbf{In}^a\mathbf{Ex}^*$.
3. $\mathbf{Mul}^j_k\mathbf{Im}^a\mathbf{Ex}^0 \subset \mathbf{Mul}^j_k\mathbf{Im}^a\mathbf{Ex}^1 \subset \cdots \subset \mathbf{Mul}^j_k\mathbf{Im}^a\mathbf{Ex}^*$.

Corollary 5.26. *Let $a \in N \cup \{*\}$. Let $j, k \in N$ such that $k \geq j > \lfloor \frac{k}{2} \rfloor$. Then,*

1. $\mathbf{Mul}^j_k\mathbf{N}^0\mathbf{Ex}^a \supset \mathbf{Mul}^j_k\mathbf{N}^1\mathbf{Ex}^a \supset \cdots \supset \mathbf{Mul}^j_k\mathbf{N}^*\mathbf{Ex}^a$.
2. $\mathbf{Mul}^j_k\mathbf{In}^0\mathbf{Ex}^a \supset \mathbf{Mul}^j_k\mathbf{In}^1\mathbf{Ex}^a \supset \cdots \supset \mathbf{Mul}^j_k\mathbf{In}^*\mathbf{Ex}^a$.
3. $\mathbf{Mul}^j_k\mathbf{Im}^0\mathbf{Ex}^a \supset \mathbf{Mul}^j_k\mathbf{Im}^1\mathbf{Ex}^a \supset \cdots \supset \mathbf{Mul}^j_k\mathbf{Im}^*\mathbf{Ex}^a$.

5.4. Comparison of Different Types of Inaccuracies. Finally, we present results which shed light on how learning from one kind of inaccuracy compares with learning from another kind of inaccuracy.

First, we show a very interesting result which implies that in the context of identification from multiple inaccurate texts, noisy texts are better than incomplete texts. This parallels previous findings about identification from single inaccurate texts. The result below shows that the collections of functions that can be identified from multiple incomplete texts can also be identified from multiple noisy texts, provided the bound on the number of errors allowed in the final program and the bound on the number of inaccuracies is the same in both cases (and also provided the cardinality of multiple texts and of acceptable texts is the same in both cases).

THEOREM 5.27. *Let* $a, b \in N \cup \{*\}$, *and* $j, k \in N$, $k \geq j > \lfloor \frac{k}{2} \rfloor$. *Then,* $\mathbf{Mul}_k^j\mathbf{In}^a\mathbf{Ex}^b \subseteq \mathbf{Mul}_k^j\mathbf{N}^a\mathbf{Ex}^b$.

Proof. Let $T_1, T_2, \ldots, T_k$ be the k input texts for f, such that at least j of the texts are a-noisy for f. Note that if $f(x) = y$, then at least j of the k texts contain (x, y). Let T_i' be the text formed from T_i such that $\mathrm{content}(T_i') = \{(x, y) \mid \mathrm{card}(\{l \mid (x, y) \in \mathrm{content}(T_l)\}) \geq j \ \wedge\ \mathrm{card}(\{z \mid (x, z) \in \mathrm{content}(T_i)\}) \leq 1 \ \wedge\ (x, y) \in \mathrm{content}(T_i)\}$. Thus, we can easily obtain T_i's from T_is in the limit. Also, if T_i was an a-noisy text for f, then T_i' is an a-incomplete text for f. The theorem follows. □

As a contrast to the above result, the next result implies that the collections of functions which can be identified from multiple noisy texts cannot always be identified from a single incomplete text (and, hence from multiple incomplete texts). The result shows that there are collections of functions for which a 0-error program can be identified from any k texts, at least $j(> \lfloor \frac{k}{2} \rfloor)$ of which are only $*$-noisy, but for which even a $*$-error program cannot be identified from single 1-incomplete text.

THEOREM 5.28. *Let* $j, k \in N$, $k \geq j > \lfloor \frac{k}{2} \rfloor$. *Then,* $\mathbf{Mul}_k^j\mathbf{N}^*\mathbf{Ex} - \mathbf{In}^1\mathbf{Ex}^* \neq \emptyset$.

Proof. Follows from Theorem 10 in [11] and Theorem 5.1. □

The next two results compare identification from multiple incomplete texts and multiple imperfect texts. As expected, imperfect texts turn out to be worse for identification than incomplete texts.

The next theorem shows that for $i \in N$, there are collections of functions for which a 0-error program (best program) can be identified from any k texts, at least $j(> \lfloor \frac{k}{2} \rfloor)$ of which are $(3i - 1)$-incomplete, but for which even a $*$-error program (worst acceptable program) cannot be identified from any single text which is only $2i$-imperfect (a smaller bound on inaccuracies).

THEOREM 5.29. *Let* $j, k \in N, k \geq j > \lfloor \frac{k}{2} \rfloor$. *Then,* $(\forall i \in N)[\mathbf{Mul}_k^j\mathbf{In}^{3i-1}\mathbf{Ex} - \mathbf{Im}^{2i}\mathbf{Ex}^* \neq \emptyset]$.

Proof. Follows from Theorem 58 in [14] and Theorem 5.2. □

Our final theorem is on similar lines to the previous theorem and shows the detrimental effects of imperfect text over incomplete texts. The result says that there are collections of functions for which a 0-error program (best possible) can be identified from any k texts, at least $j(> \lfloor \frac{k}{2} \rfloor)$ of which are $*$-incomplete (worst kind of missing data), but for which even a $*$-error program (worst acceptable program) cannot be identified from any single text which is only $*$-imperfect (least harmless imperfection).

THEOREM 5.30. *Let* $j, k \in N, k \geq j > \lfloor \frac{k}{2} \rfloor$. *Then,* $\mathbf{Mul}_k^j\mathbf{In}^*\mathbf{Ex} - \mathbf{Im}^*\mathbf{Ex}^* \neq \emptyset$.

Proof. Follows from Theorem 60 in [14] and Theorem 5.2. □

6. Conclusion. We presented arguments against the idealized assumption in Gold's paradigm that a learning agent receives data from a single and accurate source. Gold's paradigm was suitably extended to account for the possibility that a learning agent may receive data from multiple sources, some of which may be inaccurate. Results were presented for the learning task of scientific inquiry modeled as identification in the limit of computer programs for computable functions from their graphs.

For each kind of inaccuracy, we established sufficient conditions when, for the same bound on the number of inaccuracies in both the multiple inaccurate texts and the single inaccurate text, and for the same bound on the number of errors allowed in the final hypothesis for the multiple and single cases, identification from multiple sources is not a restriction. We provided significant partial results for the difficult problem of determining when identification from multiple inaccurate texts is a restriction over identification from a single inaccurate text. We also demonstrated cases under which the detrimental effects of multiple inaccurate texts can be overcome by either allowing more errors in the final program or by decreasing the bound on the number of inaccuracies. We also established the usual hierarchies for each kind of inaccuracy in which

- keeping all the other parameters fixed and increasing the bound on the number of errors allowed in the final program improves learnability;
- keeping all the other parameters fixed and increasing the bound on the number of inaccuracies in the acceptable texts restricts learnability.

Finally we were able to demonstrate that learning from noisy texts is preferable to learning from incomplete texts which in turn is preferable to learning from imperfect texts.

It should be noted that several of the results presented in Section 5.2 can easily be generalized to the case in which a machine is fed k texts and for at least j of these texts the inaccuracies are acceptable. Also, results in the present manuscript are about a simple criterion of success, viz., **Ex**-identification. We can also prove corresponding results for a more general criterion of learning, viz., **Bc**-identification (see [8] for definition; also known as GN^∞ [2] in the Russian literature). Additionally, we would like to note that similar results can also be shown to hold for another learning task, viz., first language acquisition modeled as identification in the limit of grammars for recursively enumerable languages from a text of these languages. The criterion of success for language acquisition is known as **TxtEx**-identification (see [7] for definition).

Finally, we end this paper on a speculatory note pointing to future research directions. Scientific success is often not limited to the success of a single scientist receiving data from multiple, possibly inaccurate, sources. In actual practice, a number of scientists are simultaneously investigating a phenomenon, each receiving data from multiple, possibly inaccurate, sources. Scientific success is achieved if any one of these scientists is successful. This scenario could be modeled as a 'team' of learning machines, each member of the team receiving multiple inaccurate texts. The team is successful just in case at least one member of the team converges to a correct program for the function being learned (see Smith [24] for discussion of team identification from a *single* and *accurate* text). However, given our experience with the combinatorial difficulty of the subject matter of this paper, it is quite likely that a study incorporating teams and multiple inaccurate environments may turn out to be very complex, and it is not clear, at this stage, what additional insights such an investigation may provide.

Acknowledgements. We would like to thank John Case, Mark Fulk, and Errol Lloyd for encouragement and advice. Helpful discussions were provided by Sudhir Jha, Lata Narayanan, and Rajeev Raman. We would also like to thank Andrea Lobo for careful reading of an earlier draft of the paper.

During the initial stages of this investigation Ganesh Baliga was in the Department of Computer and Information Sciences at the University of Delaware; Sanjay

Jain was in the Department of Computer and Information Sciences at the the University of Delaware and at the Institute of Systems Science at the National University of Singapore; and Arun Sharma was in the Department of Brain and Cognitive Sciences at the Massachusetts Institute of Technology.

We would also like to express our gratitude to Prof. S.N. Maheshwari of the Department of Computer Science and Engineering at IIT-Delhi for making the facilities of his department available to us during the preparation of an early draft of this paper.

Finally, we would like to thank the referees for valuable comments.

REFERENCES

[1] *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory.* ACM Press, July 1992.

[2] J. M. Barzdin. Two theorems on the limiting synthesis of functions. *In Theory of Algorithms and Programs, Latvian State University, Riga*, 210:82–88, 1974. In Russian.

[3] L. Blum and M. Blum. Toward a mathematical theory of inductive inference. *Information and Control*, 28:125–155, 1975.

[4] M. Blum. A machine independent theory of the complexity of recursive functions. *Journal of the ACM*, 14:322–336, 1967.

[5] J. Case. Periodicity in generations of automata. *Mathematical Systems Theory*, 8:15–32, 1974.

[6] J. Case. Infinitary self-reference in learning theory. *Journal of Experimental and Theoretical Artificial Intelligence*, 4:281–293, 1993. To appear in the special issue from AII 92.

[7] J. Case and C. Lynes. Machine inductive inference and language identification. In M. Nielsen and E. M. Schmidt, editors, *Proceedings of the 9th International Colloquium on Automata, Languages and Programming*, pages 107–115. Springer-Verlag, 1982. Lecture Notes in Computer Science 140.

[8] J. Case and C. Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science*, 25:193–220, 1983.

[9] M. Fulk. *A Study of Inductive Inference Machines.* PhD thesis, SUNY at Buffalo, 1985.

[10] M. Fulk and J. Case (editors). *Proceedings of the Third Annual Workshop on Computational Learning Theory.* Morgan Kaufmann Publishers, Inc., August 1990.

[11] M. A. Fulk and S. Jain. Learning in the presence of inaccurate information. In R. Rivest, D. Haussler, and M. K. Warmuth, editors, *Proceedings of the Second Annual Workshop on Computational Learning Theory, Santa Cruz, California*, pages 175–188. Morgan Kaufmann Publishers, Inc., August 1989.

[12] E. M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[13] D. Haussler and L. Pitt˜(editors). *Proceedings of the 1988 Workshop on Computational Learning Theory.* Morgan Kaufmann Publishers, Inc., August 1988.

[14] S. Jain. *Learning in the Presence of Additional Information and Inaccurate Information.* PhD thesis, University of Rochester, 1990.

[15] S. Jain. Program synthesis in the presence of infinite number of inaccuracies. In S.Arikawa and K. P. Jantke, editors, *Proceedings of the Fifth International Workshop on Algorithmic Learning Theory, Reinhardsbrunn Castle, Germany*, pages 333–348, October 1994.

[16] M. Machtey and P. Young. *An Introduction to the General Theory of Algorithms.* North Holland, New York, 1978.

[17] D. Osherson, M. Stob, and S. Weinstein. *Systems that Learn, An Introduction to Learning Theory for Cognitive and Computer Scientists.* MIT Press, Cambridge, Mass., 1986.

[18] D. Osherson and S. Weinstein. A note on formal learning theory. *Cognition*, 11:77–88, 1982.

[19] H. Putnam. Reductionism and the nature of psychology. *Cognition*, 2:131–146, 1973.

[20] R. Rivest, D. Haussler, and M. K. Warmuth (editors). *Proceedings of the Second Annual Workshop on Computational Learning Theory.* Morgan Kaufmann Publishers, Inc., August 1989.

[21] H. Rogers. Gödel numberings of partial recursive functions. *Journal of Symbolic Logic*, 23:331–341, 1958.

[22] H. Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw Hill, New York, 1967. Reprinted, MIT Press 1987.

[23] G. Schäfer-Richter. Some results in the theory of effective program synthesis - learning by defective information. *Lecture Notes in Computer Science*, 215:219–225, 1986.

[24] C. Smith. The power of pluralism for automatic program synthesis. *Journal of the ACM*, 29:1144–1165, 1982.

[25] R. J. Solomonoff. A formal theory of inductive inference, Part I. *Information and Control*, 7:1–22, 1964.

[26] R. J. Solomonoff. A formal theory of inductive inference, Part II. *Information and Control*, 7:224–254, 1964.

[27] L. G. Valiant and M. K. Warmuth (editors). *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*. Morgan Kaufmann Publishers, Inc., August 1991.