# Learning with Multiple Correct Answers - Regret Bounds under Different Feedback Models

**Alireza F. Pour** *
David Cheriton School of Computer Science
University of Waterloo and Vector Institute
alireza.fathollahpour@uwaterloo.ca

**Farnam Mansouri***
David Cheriton School of Computer Science
University of Waterloo and Vector Institute
f5mansou@uwaterloo.ca

**Shai Ben-David**
David Cheriton School of Computer Science
University of Waterloo and Vector Institute
shai@uwaterloo.ca

## Abstract

We study the problem of learning with multiple correct answers, where each instance admits a set of valid labels. We primarily focus on the online setup, where in each round the learner must output a valid label for the queried example. This setting is motivated by language generation, in which a prompt may admit many acceptable completions, but not every completion is acceptable. We study this problem under three feedback models. For each model, we characterize the optimal mistake bound in the realizable setting using an appropriate combinatorial dimension. We then show that the rate of regret can be constant, linear, or sublinear across the three models in the agnostic setting. Our results also imply sample complexity bounds for the batch setup that depend on the respective combinatorial dimensions.

## 1 Introduction

Many modern prediction tasks admit multiple correct outputs for a single input. A prominent example is language generation, where a prompt may have many acceptable completions but not all completions are acceptable. This phenomenon arises in settings such as open-ended question answering, dialogue generation, and instruction following, where correctness is better viewed as membership in a set of valid responses rather than by a single target label [LZ07; LMS+17; OWJ+22]. Similar considerations arise in recommendation systems and information retrieval, where multiple outputs may equally satisfy the task requirements [Joa02; Liu+09]. These settings call for learning formulations in which correctness is inherently set-valued rather than single-valued.

In this work we analyze and compare learning with multiple correct answers with respect to three aspects; Online vs Batch learning, Realizable vs Agnostic setups, and the type of feedback available to the learner. We focus mainly on the online setting. An online problem proceeds in rounds. In each round, the learner is presented with an example and its objective is to produce a correct answer for that example. This setup naturally accommodates the following feedback models for the learner:

- *Mistake-unknown feedback*, where the learner observes only a single correct label in each round (as studied by Joshi et al. [JLB+25]);
- *Mistake-known feedback*, where, in addition to a single correct label, the learner also observes whether its prediction was correct; and

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

- *Set-valued feedback*, where the full set of correct labels is revealed to the learner (as studied by Raman et al. [RST24]).[2]

The three feedback models represent progressively more informative forms of supervision. For example, in language generation, training data often consist of prompts paired with a single demonstrated completion. During interaction with users, a model may additionally receive feedback indicating whether a generated completion is acceptable, corresponding to the mistake-known model. Set-valued feedback, which reveals the entire set of correct labels, represents the most informative model. As we show, these differences in the information available to the learner can fundamentally change the difficulty of learning.

Our goal is to develop a unified theory of learning with multiple correct answers under these feedback models. We consider classes of multi-label hypotheses, that is, functions $h : \mathcal{X} \to 2^{\mathcal{Y}}$ that assign a set of labels to each instance. In the realizable setting, we characterize the optimal mistake bounds through three feedback-dependent Littlestone dimensions. In the agnostic setting, we show a stronger separation: mistake-unknown feedback can lead to linear regret even for classes with small realizable mistake bounds, while observing the mistake bit guarantees sublinear regret for classes with finite $\mathsf{LDknown}$, and set-valued feedback can even yield constant regret. This contrasts with single-label learning, where the three feedback models coincide [BPS09; HMR+23]. Our results further imply sample complexity bounds for batch learning.

**Our Contributions.** We formalize learning with multiple correct answers in the online setting and under the three models of feedback (Section 2).

- **Realizable online learning.** We introduce novel combinatorial parameters $\mathsf{LDunknown}$, $\mathsf{LDknown}$, and $\mathsf{LDset}$ and based on these parameters we establish the first characterization of realizable online learnability under the three feedback models (Section 3). The mistake-unknown setting was studied by Joshi et al. [JLB+25], but only for finite hypothesis classes. Our characterization extends their result from finite (only) to any hypothesis classes.
- **Agnostic online learning.** We provide examples of three different regret rates across the three feedback models: linear in $T$, $O(\sqrt{T\log(T)})$, and constant.
  - **Mistake-unknown.** We show that in the agnostic setting, the regret of mistake-unknown multi-label learning can grow linearly in the number of rounds $T$, even for a hypothesis class of size 3 with $\mathsf{LDunknown} = 1$ (Section 4.1). *This highlights a distinction between realizable and agnostic learning: characterizations of the realizable setting fail to capture any meaningful regret bounds in the agnostic setting. This is in contrast to many learning settings where realizable learning implies agnostic learning [HKLM22; BPS09].*
  - **Mistake-known.** We prove a novel upper bound of $O\left(\sqrt{\mathsf{LDknown}(\mathcal{H})|\mathcal{Y}|T\log(T)}\right)$ on the regret of agnostic mistake-known multi-label learning of any class (Theorem 4.2). This is only $\tilde{O}(\sqrt{|\mathcal{Y}|\log(T)})$ away from known lower bounds for single-label setting which also apply here [DSBS15].
  - **Set-valued.** We show that the regret of agnostic set-valued multi-label learning for a variety of classes with $\mathsf{LDset} \geq 1$ (including the hypothesis class of size 3 mentioned above) admits a constant upper bound, and we characterize all finite classes admitting such a bound (Theorem 4.4). We also attain a $O\left(\sqrt{\mathsf{LDset}(\mathcal{H})T\log(T)}\right)$ upper bound for general hypothesis classes (Section 4.3). *This variety of regret rates stands in contrast to single-valued learning where* any *hypothesis class must have a regret of order* $\Omega(\sqrt{T})$ *[DSBS15].*
- **Batch learning.** *Our results yield novel sample-complexity upper bounds for the corresponding batch learning problems. We get the first known upper bound for the loss for batch learning in the full-set feedback model. We also improve the result of Joshi et al. [JLB+25] for the case of single correct answer from finite classes to any (possibly infinite) classes.* Due to space constraints, we defer the formal setup and our results for the batch problem to Appendix A.

**Remark 1.1.** *For the specific case of single-label class $\mathcal{H}$, all the three feedback models will be equivalent to the multiclass online learning, where the mistake bound is characterized by the Littlestone dimension,* $\mathsf{LD}(\mathcal{H})$ *[DSBS15; HMR+23], and therefore,* $\mathsf{LDunknown}(\mathcal{H}) = \mathsf{LDknown}(\mathcal{H}) =$

[2] One may also consider a fourth model of feedback, where the learner only gets a single bit indicating whether a mistake has been made. Such feedback is very weak since the learner can be forced to err on a single instance multiple times depending on the size of the label space. We therefore chose not to discuss this model here.

$\mathsf{LDset}(\mathcal{H}) = \mathsf{LD}(\mathcal{H})$. *In the multiclass setting, the regret is* $\tilde{\Theta}(\sqrt{\mathsf{LD}(\mathcal{H})T})$ *while we show the regret of a multi-label class can be independent of the corresponding Littlestone dimensions. Our work therefore suggests that the landscape of agnostic multi-label learning has a rich structure and may require combinatorial objects beyond those that characterize realizable learnability.*

## 2 Learning Setup

Let $\mathcal{X}$ be a domain set, $\mathcal{Y}$ a finite set of labels and $T$ a natural number. In the online multi-label learning, an adversary and a learner play a sequential game for $T$ rounds. In each round $t \in [T]$, the adversary selects an instance $x_t \in \mathcal{X}$ and a set of acceptable labels $S_t \subseteq \mathcal{Y}$ and reveals $x_t$ to the learner. The learner makes a (potentially randomized) prediction $\hat{y}_t \in \mathcal{Y}$ and suffers loss $\mathbb{1}[\hat{y}_t \notin S_t]$. The learner's goal is to minimize its cumulative loss over the $T$ rounds.

We consider the following three feedback models: for each round $t \in [T]$,

- **(Mistake-unknown).** In addition to choosing $(x_t, S_t)$, the adversary also selects a predetermined label $y_t \in S_t$ (before observing $\hat{y}_t$) and reveals it to the learner.
- **(Mistake-known).** In addition to revealing the predetermined label $y_t$, the adversary also reveals a binary indicator specifying whether the learner made a mistake, namely $\mathbb{1}[\hat{y}_t \notin S_t]$.
- **(Set-valued).** The adversary reveals the entire set $S_t$.

In all cases, the feedback is revealed only after the learner has chosen $\hat{y}_t$.

A multi-label hypothesis class $\mathcal{H}$ is a set of functions mapping $\mathcal{X}$ to $2^{\mathcal{Y}}$. Denote by $\Pi(\mathcal{Y})$ the set of probability measures on $\mathcal{Y}$. An online learner $\mathcal{A} : (\mathcal{X} \times \mathcal{Z})^* \to \Pi(\mathcal{Y})^{\mathcal{X}}$ is a mapping that each time $t$, gets as input the past sequence of instances $x_1, \ldots, x_{t-1}$ and feedbacks $z_1, \ldots, z_{t-1}$ along with a new test point $x_t$ and outputs a random variable $p_{\mathcal{A}}(t) := \mathcal{A}(x_1, z_1, \ldots, x_{t-1}, z_{t-1})(x_t)$ over $\mathcal{Y}$. Depending on the feedback model, $\mathcal{Z}$ is always chosen as $\mathcal{Y}$, $\mathcal{Y} \times \{0, 1\}$, or $2^{\mathcal{Y}}$.

We say that the multi-label learning in the online setting satisfies the realizability assumption if there exists some $h^* \in \mathcal{H}$ such that for all $t \in [T]$ we have $h^*(x_t) = S_t$. Our primary focus in the realizable setting is the mistake bound (i.e., the cumulative loss) of the learner. Throughout our analysis of the realizable online learning, we restrict attention to deterministic learning algorithms.

We now give the definitions of mistake bound for the three feedback models. In the mistake-unknown setting, we have $\mathcal{Z} = \mathcal{Y}$ and, for any $t \in [T]$, the feedback $z_t$ is a (single) label $y_t \in S_t$.

**Definition 2.1** (Unknown Mistake Bound)**.** *Unknown mistake bound of a deterministic learning algorithm* $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^* \to \mathcal{Y}^{\mathcal{X}}$ *with respect to a hypothesis class* $\mathcal{H}$ *and length* $T \in \mathbb{N}$ *is defined as*

$$\mathrm{M}_{\mathcal{A}}^{\mathrm{unknown}}(T, \mathcal{H}) := \sup_{\substack{h^* \in \mathcal{H}, (x_1, y_1), \ldots, (x_T, y_T): \\ y_t \in h^*(x_t)}} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left[p_{\mathcal{A}}(t) \notin h^*(x_t)\right]\right]. \tag{1}$$

In the mistake-known setting, where $\mathcal{Z} = \mathcal{Y} \times \{0, 1\}$, the learner also observes as feedback whether it made a mistake. Formally, $\forall t \in [T]$, $z_t = (y_t, b_t)$, where $b_t := \mathbb{1}[p_{\mathcal{A}}(t) \notin S_t]$ is the mistake bit.

**Definition 2.2** (Known Mistake Bound)**.** *Known mistake bound of a deterministic learning algorithm* $\mathcal{A} : (\mathcal{X} \times \mathcal{Y} \times \{0, 1\})^* \to \mathcal{Y}^{\mathcal{X}}$ *with respect to hypothesis class* $\mathcal{H}$ *and length* $T \in \mathbb{N}$ *is defined as*

$$\mathrm{M}_{\mathcal{A}}^{\mathrm{known}}(T, \mathcal{H}) := \sup_{\substack{h^* \in \mathcal{H}, (x_1, y_1), \ldots, (x_T, y_T): \\ y_t \in h^*(x_t)}} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left[p_{\mathcal{A}}(t) \notin h^*(x_t)\right]\right]. \tag{2}$$

Finally, in the set-valued domain, the learner is given full information. Particularly, we have $\mathcal{Z} = 2^{\mathcal{Y}}$ and at each time $t \in [T]$, $z_t = S_t$ and the entire label set is revealed.

**Definition 2.3** (Set-Valued Mistake Bound)**.** *Set-valued mistake bound of a deterministic learning algorithm* $\mathcal{A} : (\mathcal{X} \times 2^{\mathcal{Y}})^* \to \mathcal{Y}^{\mathcal{X}}$ *with respect to a hypothesis class* $\mathcal{H}$ *and length* $T \in \mathbb{N}$ *is defined as*

$$\mathrm{M}_{\mathcal{A}}^{\mathrm{set}}(T, \mathcal{H}) := \sup_{h^* \in \mathcal{H}, (x_1, h^*(x_1)), \ldots, (x_T, h^*(x_T))} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left[p_{\mathcal{A}}(t) \notin h^*(x_t)\right]\right]. \tag{3}$$

We will also consider the agnostic online multi-label learning. In this setting, we will define the *regret* of an online algorithm after $T$ rounds under the three different feedback models we consider.

**Definition 2.4** (Mistake-Unknown Regret)**.** *Mistake-unknown regret of a learning algorithm* $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^* \to \Pi(\mathcal{Y})^{\mathcal{X}}$ *with respect to a multi-label hypothesis class* $\mathcal{H}$ *and length* $T \in \mathbb{N}$ *is defined as*

$$\mathrm{R}_{\mathcal{A}}^{\text{unknown}}(T, \mathcal{H}) := \sup_{\substack{(x_1,y_1,S_1),\ldots,(x_T,y_T,S_T):\\ y_t \in S_t}} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}[p_{\mathcal{A}}(t) \notin S_t] - \inf_{h\in\mathcal{H}} \sum_{t=1}^{T} \mathbb{1}\left[h(x_t) \neq S_t\right]\right]. \quad (4)$$

**Definition 2.5** (Mistake-Known Regret)**.** *Mistake-known regret of a learning algorithm* $\mathcal{A} : (\mathcal{X} \times \mathcal{Y} \times \{0,1\})^* \to \Pi(\mathcal{Y})^{\mathcal{X}}$ *with respect to a multi-label hypothesis class* $\mathcal{H}$ *and length* $T \in \mathbb{N}$ *is defined as*

$$\mathrm{R}_{\mathcal{A}}^{\text{known}}(T, \mathcal{H}) := \sup_{\substack{(x_1,y_1,S_1),\ldots,(x_T,y_T,S_T):\\ y_t \in S_t}} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left[p_{\mathcal{A}}(t) \notin S_t\right] - \inf_{h\in\mathcal{H}} \sum_{t=1}^{T} \mathbb{1}\left[h(x_t) \neq S_t\right]\right]. \quad (5)$$

**Definition 2.6** (Set-Valued Regret)**.** *Set-valued regret of a learning algorithm* $\mathcal{A} : (\mathcal{X} \times 2^{\mathcal{Y}})^* \to \Pi(\mathcal{Y})^{\mathcal{X}}$ *with respect to a multi-label hypothesis class* $\mathcal{H}$ *and length* $T \in \mathbb{N}$ *is defined as*

$$\mathrm{R}_{\mathcal{A}}^{\text{set}}(T, \mathcal{H}) := \sup_{(x_1,S_1),\ldots,(x_T,S_T)} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left[p_{\mathcal{A}}(t) \notin S_t\right] - \inf_{h\in\mathcal{H}} \sum_{t=1}^{T} \mathbb{1}\left[h(x_t) \neq S_t\right]\right]. \quad (6)$$

**Notations.** For any $U = (u_i)_{i=1}^{m}$ and $t \in [m]$ we denote $U_{<t} := (u_i)_{i=1}^{t-1}$, and $U_{\leq t} := (u_i)_{i=1}^{t}$.

### 2.1 A Comparison Between Notions of Regret

Joshi et al. [JLB+25] introduce the following notion of regret for online mistake-unknown multi-label learning with respect to a multi-label hypothesis class $\mathcal{H}$:

$$\sup_{(x_1,y_1),\ldots,(x_T,y_T)} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}[p_{\mathcal{A}}(t) \notin h^*(x_t)] - \sum_{t=1}^{T} \mathbb{1}[y_t \notin h^*(x_t)]\right], \quad (7)$$

where $h^* \in \arg\min_{h\in\mathcal{H}} \sum_{t=1}^{T} \mathbb{1}[y_t \notin h(x_t)]$. Observe, however, that the term $\mathbb{1}[p_{\mathcal{A}}(t) \notin h^*(x_t)]$ measures the error of the algorithm relative to the hypothesis that contains the correct label more often—even though such a hypothesis may also include many incorrect labels. Thus, competing with such a hypothesis is not necessarily well aligned with the underlying prediction task. Instead, we are interested in evaluating hypotheses with respect to the underlying set of acceptable labels, which motivates measuring the learner's error against the true loss sequence, i.e., $\mathbb{1}[p_{\mathcal{A}}(t) \notin S_t]$, as in Definition 2.4. This definition also extends naturally to the mistake-known and set-valued settings while notions defined as in Equation (7) do not admit such a direct generalization.

Furthermore, for a single-label hypothesis class $\mathcal{C} \subseteq \mathcal{Y}^{\mathcal{X}}$, Raman et al. [RST24] introduced the following notion of regret for online set-valued multi-label learning relative to $\mathcal{C}$:

$$\sup_{(x_1,S_1),\ldots,(x_T,S_T)} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}[p_{\mathcal{A}}(t) \notin S_t] - \inf_{c\in\mathcal{C}} \sum_{t=1}^{T} \mathbb{1}[c(x_t) \notin S_t]\right].$$

In this formulation, while the feedback is set-valued, the hypothesis class itself remains single-labeled. When multiple labels are acceptable for a given instance, it is natural to consider hypothesis classes that may themselves assign multiple correct labels, as in our setup.

At first glance, an alternative formulation of regret to Definition 2.4 would be

$$\sup_{\substack{(x_1,y_1,S_1),\ldots,(x_T,y_T,S_T):\\ y_t \in S_t}} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}[p_{\mathcal{A}}(t) \notin S_t] \; - \; \inf_{h\in\mathcal{H}} \sum_{t=1}^{T} \mathbb{1}[h(x_t) \not\subseteq S_t]\right]. \quad (8)$$

The principle of replacing "not equal ($\neq$) " with "not a subset of ($\not\subseteq$)" can be applied to the mistake-known and set-valued feedback models too. The following example shows that this notion of regret makes the adversary overly powerful and can render the feedback $y_t$ completely non-informative.

**Example 1.** *Consider the hypothesis class $\mathcal{H} = \{h_1, h_2\}$ over the domain $\mathcal{X} = \{x\}$ and label set $\mathcal{Y} = [100]$, where $h_1(x) = \{1\}$ and $h_2(x) = \{2\}$. Let $h^* = h_b$ where $b$ is chosen uniformly at random from $\{1, 2\}$. In each round $t$, we select $r$ uniformly at random from $\{3, \ldots, 100\}$ and set $S_t = \{b, r\}, y_t = r$. We have*

$$\inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{1}[h(x_t) \nsubseteq S_t] = 0.$$

*Here, the feedback $y_t$ is completely non-informative under both the mistake-known and mistake-unknown models. In particular, in the mistake-unknown model, the probability that $b = 1$ is $1/2$ and the probability of selecting any particular $r \in [3, 100]$ is $1/98$. Consequently, the notion of regret defined in (8) is always at least $T/2$.*

## 3 Combinatorial Parameters and Realizable Online Multi-Label Learning

We introduce combinatorial dimensions for each feedback model which are defined via trees that reflect the game between the learner and an adversary. We note that the dimension we introduce for the set-valued feedback model (Definition 3.1) closely matches the set Littlestone dimension studied by Raman et al. [RST24].

A set-valued tree $\mathcal{T}$ is a complete $\mathcal{X} \times 2^{\mathcal{Y}}$ valued, $\mathcal{Y}-$arry tree. Any path originating from root in this tree is defined by a sequence $\sigma = (\hat{y}_1, ..., \hat{y}_t) \in \mathcal{Y}^t$. Denote by $\nu(\sigma) = (\nu_x(\sigma), \nu_S(\sigma))$ the final node in path $\sigma$.

**Definition 3.1** (Set-Valued Littlestone Dimension)**.** *Let $\mathcal{T}$ be a complete set-valued tree with depth $d$. We say $\mathcal{T}$ is shattered by $\mathcal{H}$ if, for any path $\sigma = (\hat{y}_1, ..., \hat{y}_d)$ from root to some leaf, there exists a $h_\sigma \in \mathcal{H}$ such that for each $t \in [d]$, we have*

*(i).* Consistency: $\nu_S(\sigma_{\leq t}) = h_\sigma\big(\nu_x(\sigma_{<t})\big)$,
*(ii).* Mistake requirement: $\hat{y}_t \notin h_\sigma(\nu_x(\sigma_{<t}))$.

*The Littlestone dimension* LDset *of $\mathcal{H}$ in the set-valued online setting is then defined as the depth of the largest shattered tree.*

It is worth observing that in the above definition, $\nu_x(\sigma_{<t})$ is the unlabeled instance sitting at the parent of node $\nu(\sigma_{\leq t})$.

A multi-label tree $\mathcal{T}$ is a $\mathcal{X} \times \mathcal{Y}$ valued, $\mathcal{Y}-$arry tree such that each node either has no children (a leaf) or $|\mathcal{Y}|$ children labeled for all $y \in \mathcal{Y}$. Any path originating from root in this tree is defined by a sequence $\sigma = (\hat{y}_1, ..., \hat{y}_t) \in \mathcal{Y}^t$. Denote $\nu(\sigma) = (\nu_x(\sigma), \nu_y(\sigma))$ to be the final node in path $\sigma$.

**Definition 3.2** (Mistake-Known Multi-Label Littlestone Dimension)**.** *Let $\mathcal{T}$ be a complete multi-label tree with depth $d$. We say $\mathcal{T}$ is shattered by $\mathcal{H}$ if, for any path $\sigma = (\hat{y}_1, ..., \hat{y}_d)$ from root to some leaf, there exists a $h_\sigma \in \mathcal{H}$ such that for each $t \in [d]$, we have*

*(i)* Consistency: $\nu_y(\sigma_{\leq t}) \in h_\sigma\big(\nu_x(\sigma_{<t})\big)$,
*(ii)* Mistake requirement: $\hat{y}_t \notin h_\sigma(\nu_x(\sigma_{<t}))$.

*The Littlestone dimension* LDknown *of $\mathcal{H}$ in the mistake-known online setting is then defined as the depth of the largest shattered tree.*

**Definition 3.3** (Mistake-Unknown Multi-Label Littlestone Dimension)**.** *Let $\mathcal{T}$ be a multi-label tree and let* off $: \mathcal{H} \to \mathbb{N} \cup \{0\}$ *be a* mistake offset function*. We say that $\mathcal{T}$ is* $d$-shattered *by $(\mathcal{H}, \mathsf{off})$ if for every root-to-leaf path $\sigma = (\hat{y}_1, \ldots, \hat{y}_T)$ in $\mathcal{T}$, there exists a hypothesis $h_\sigma \in \mathcal{H}$ such that:*

*(i)* Consistency: *for all* $t \in [T]$, $\nu_y(\sigma_{\leq t}) \in h_\sigma\big(\nu_x(\sigma_{<t})\big)$,
*(ii)* Cumulative mistake requirement: $\sum_{t=1}^{T} \mathbb{1}\left[\hat{y}_t \notin h_\sigma(\nu_x(\sigma_{<t}))\right] + \mathsf{off}(h_\sigma) \;\geq\; d$.

*Specifically for any tree $\mathcal{T}$ with depth $0$, we say $\mathcal{T}$ is $d$-shattered by $(\mathcal{H}, \mathsf{off})$ if there exists $h_\emptyset \in \mathcal{H}$ such that $\mathsf{off}(h_\emptyset) \geq d$. The* mistake-unknown multi-label Littlestone dimension $\mathsf{LDunknown}(\mathcal{H}, \mathsf{off})$ *is the largest integer $d \in \mathbb{N}$ such that there exists a $d$-shattered tree by $(\mathcal{H}, \mathsf{off})$. Moreover, we denote $\mathsf{LDunknown}(\mathcal{H}) := \mathsf{LDunknown}(\mathcal{H}, \mathbf{0})$, where $\mathbf{0}(h) = 0$ for all $h \in \mathcal{H}$.*

Intuitively, $\mathsf{off}(h)$ accounts for the number of mistakes already incurred by hypothesis $h$, reducing the remaining mistake budget available along any path.

**Claim 3.4.** *For any multi-label hypothesis class $\mathcal{H}$ and any offset function $\mathsf{off} : \mathcal{H} \to \mathbb{N}$, we have* $\mathsf{LDunknown}(\mathcal{H}, \mathsf{off}) \geq \sup_{h \in \mathcal{H}} \mathsf{off}(h)$.

*Proof.* Take the tree $\mathcal{T}$ of depth 0 that only has a root, which is also a leaf. It is obvious that $h^\star$ with $\mathsf{off}(h^\star) = \sup_{h \in \mathcal{H}} \mathsf{off}(h)$ satisfies property (ii) of Definition 3.3 for the path $\sigma = \emptyset$. This concludes that $(\mathcal{H}, \mathsf{off})$ will $\sup_{h \in \mathcal{H}} \mathsf{off}(h)$-shatter $T$. □

The following theorem characterizes when all three multi-label online dimensions collapse to zero. The proof is deferred to Appendix C.1.

**Theorem 3.5.** *For any multi-label hypothesis class $\mathcal{H}$, we have*

$$\mathsf{LDset}(\mathcal{H}) = 0 \;\Leftrightarrow\; \mathsf{LDknown}(\mathcal{H}) = 0 \;\Leftrightarrow\; \mathsf{LDunknown}(\mathcal{H}) = 0 \;\Leftrightarrow\; \forall x \in \mathcal{X} : \bigcap_{h \in \mathcal{H}} h(x) \neq \emptyset.$$

The next result shows that the gap between the three dimensions in Corollary 3.8 can be arbitrarily large. Moreover, it shows that they behave differently under unions. Its proof appears in Appendix C.3.

**Theorem 3.6.** *For every $n \geq 1$, there exists a multi-label hypothesis class $\mathcal{H}_n$ over a finite label space such that*

$$\mathsf{LDset}(\mathcal{H}_n) = 1, \qquad \mathsf{LDknown}(\mathcal{H}_n) = n, \qquad \mathsf{LDunknown}(\mathcal{H}_n) = \infty.$$

*Moreover, for $\mathsf{LD} \in \{\mathsf{LDset}, \mathsf{LDknown}\}$ and any $\mathcal{H}_1, \mathcal{H}_2$, $\mathsf{LD}(\mathcal{H}_1 \cup \mathcal{H}_2) \leq \mathsf{LD}(\mathcal{H}_1) + \mathsf{LD}(\mathcal{H}_2) + 1$. This inequality does not hold for $\mathsf{LDunknown}$, even when $\mathsf{LDunknown}(\mathcal{H}_1) = \mathsf{LDunknown}(\mathcal{H}_2) = 0$.*

### 3.1 A Characterization of Realizable Online Multi-Label Learning

The following theorem characterizes the mistake bound of deterministic learners in the realizable online multi-label learning problem and under different feedback models, using the Littlestone dimensions defined in Section 3. We define a Standard Optimal Algorithm (SOA) for each setup and show they achieve optimal mistake bounds. The proof of Theorem 3.7 and the description of SOAs appear in Appendix C.2.

**Theorem 3.7.** *For any multi-label hypothesis class $\mathcal{H}$, and any $T \in \mathbb{N}$ we have the following characterizations of the optimal mistake bound for the three feedback models:*

$$\begin{aligned}
&\textit{Mistake-unknown feedback:} && \inf_{\mathcal{A}} \mathrm{M}^{\mathrm{unknown}}_{\mathcal{A}}(T, \mathcal{H}) = \min(\mathsf{LDunknown}(\mathcal{H}), T).\\
&\textit{Mistake-known feedback:} && \inf_{\mathcal{A}} \mathrm{M}^{\mathrm{known}}_{\mathcal{A}}(T, \mathcal{H}) = \min(\mathsf{LDknown}(\mathcal{H}), T).\\
&\textit{Set-valued feedback:} && \inf_{\mathcal{A}} \mathrm{M}^{\mathrm{set}}_{\mathcal{A}}(T, \mathcal{H}) = \min(\mathsf{LDset}(\mathcal{H}), T).
\end{aligned}$$

Joshi et al. [JLB+25] showed that for any finite concept class $\mathcal{H}$, there exists a deterministic algorithm with a mistake bound of at most $\log_2 |\mathcal{H}|$ in the mistake-unknown setup. Combined with Theorem 3.7, this immediately yields the following.

**Corollary 3.8.** *For a finite multi-label hypothesis class $\mathcal{H}$ we have* $\mathsf{LDset}(\mathcal{H}) \leq \mathsf{LDknown}(\mathcal{H}) \leq \mathsf{LDunknown}(\mathcal{H}) \leq \log_2 |\mathcal{H}|$.

## 4 Agnostic Online Multi-label Learning

Our first result in this section shows a separation between realizable and agnostic online learnability in the mistake-unknown setting by exhibiting a simple hypothesis class with $\mathsf{LDunknown}$ of 1 but a regret that always grows linearly in $T$.

In contrast, we prove for the same class that the regret under the set-valued model is only constant. This leads to the question of whether agnostic online learning is possible in the mistake-known setting which lies between the two in terms of the feedback.

Interestingly, we find that although the feedback is partial in the mistake-known setting, the regret is always sub-linear and is upper bounded by $O(\sqrt{\mathsf{LDknown}(\mathcal{H})|\mathcal{Y}|T\log(T)})$ (Theorem 4.2). This

shows a strong gap between the mistake-unknown and mistake-known feedback models in terms of whether the corresponding Littlestone dimensions yield sub-linear regret bounds in the agnostic setting. Finally, we prove that regret in the set-valued setting is bounded by $O(\sqrt{\mathsf{LDset}(\mathcal{H})T\log(T)})$ and independent of $|\mathcal{Y}|$ (Theorem 4.5), thus, holding for infinite label spaces too.

### 4.1 A Gentle Start: An Example Hypothesis Class

We start by presenting a simple hypothesis class over a singleton domain with label set $\{1,2,3\}$ for which any algorithm suffers linear regret in the mistake-unknown setting, while the regret for the same class can be made constant in the set-valued setting. We give a high-level description of the strategy that is used by the adversary to force a linear regret. The adversary selects $y_t$ uniformly at random from $\mathcal{Y}$ for all $t \in [T]$. Since there is only a single instance, in the mistake-unknown setting the learner is only a function of $(y_i)_{i<t}$ and is independent of $(S_i)_{i<t}$, as they are not revealed. Therefore, by using the label chosen most often by the learner over the $T$ rounds, the adversary can construct $S_t$ so as to force a regret linear in $T$.

The same strategy, however, does not work in the mistake-known and set-valued setups, as the prediction of the learner at time $t$ also depends on the choice of $S_i$ for $i < t$. From Theorem 4.2, we know that there is an algorithm with sub-linear regret for this class in the mistake-known setting. More interestingly, we give an algorithm (Algorithm 4 in Appendix D.1) that enjoys a constant regret of at most $\log_2(3)$ under the set-valued setting. This algorithm is inspired by Algorithm 1 in Joshi et al. [JLB+25] for mistake-unknown online learning over finite hypothesis classes.

Algorithm 4 assigns each hypothesis $h \in \mathcal{H}$ a weight $w^{(t)}(h)$, initialized to 1. At each round $t$, it predicts a label $y$ with probability proportional to the total weight of hypotheses containing $y$. It then updates the weights as follows: if a mistake occurs, it doubles the weights of hypotheses consistent with $S_t$, i.e., those relative to which regret increases, and keeps the weight of others unchanged; otherwise, it halves the weights of those that are not consistent with $S_t$, i.e., hypotheses relative to which regret decreases, and again, keeps the remaining weights unchanged.

We can then show that (i) the expectation of the sum of weights over all hypotheses is non-increasing, and (ii) that the expected sum of weights at round $T$ must have scaled exponentially with the regret.

The key differences between Algorithm 4 and that of Joshi et al. [JLB+25] are threefold: (i) our algorithm incorporates full knowledge of the true label set $S_t$; (ii) in contrast to their method, ours is probabilistic; and (iii) while they show the total sum of weights is non-increasing, we show its expectation is non-increasing. The formal proof appears in Appendix D.1.

**Theorem 4.1.** *Consider the hypothesis class $\mathcal{H}_3 = \{h_1, h_2, h_3\}$ over domain $\mathcal{X} = \{x\}$ and label set $\mathcal{Y} = \{1,2,3\}$, where $h_1(x) = \{2,3\}$, $h_2(x) = \{1,3\}$, and $h_3(x) = \{1,2\}$. We have $\mathsf{LDunknown}(\mathcal{H}_3) = \mathsf{LDset}(\mathcal{H}_3) = 1$, whereas*

$$\inf_{\mathcal{A}} \mathrm{R}^{\mathrm{unknown}}_{\mathcal{A}}(T, \mathcal{H}_3) \geq \frac{T}{9}, \quad \inf_{\mathcal{A}} \mathrm{R}^{\mathrm{set}}_{\mathcal{A}}(T, \mathcal{H}_3) \leq \log_2(3).$$

### 4.2 Agnostic Online Learning Under Mistake-Known Feedback Model

In this section we show that the regret in the mistake-known feedback model is always sub-linear by constructing a learner that attains the regret guarantee stated in Theorem 4.2. Existing results for multi-class agnostic online learning imply a $\Omega(\sqrt{T\,\mathsf{LDknown}})$ lower bound for hypothesis classes with single correct label [DSBS15].

**Theorem 4.2.** *For any multi-label hypothesis class $\mathcal{H}$, and any $T \geq \mathsf{LDknown}(\mathcal{H})$,*

$$\inf_{\mathcal{A}} \mathrm{R}^{\mathrm{known}}_{\mathcal{A}}(\mathcal{H}, T) \leq O\left(\sqrt{\mathsf{LDknown}(\mathcal{H})|\mathcal{Y}|T\log(T)}\right).$$

To prove this theorem we will use the idea of reduction to realizable learning [HMR+23]. For any subset $I \subseteq [T]$ of size at most $\mathsf{LDknown}(\mathcal{H})$, we create a mistake-known SOA (Algorithm 2 in Appendix C.2) that makes a prediction solely based on the instances in $I$ observed so far, i.e., the memory of SOA will only contain instances from $I$. We aggregate the SOAs over all such subsets to create a set of experts that advise prediction. Unlike the setting of prediction with expert advice,

the learner knows an expert made a mistake only if it follows its advice. However, executing the mistake-known SOA requires knowing whether its predictions were mistakes in every past round, which is unavailable in rounds where its recommended label is not predicted by the learner.

The above may render the suggested approach impossible. The good news is that when the problem is reduced to realizable learning, we only rely on the fact that there exists a subset $I^*$ of size at most $\mathsf{LDknown}$ from the realizable subset of data with the following property: $I^*$ exactly equals the set of mistakes that the SOA defined on $I^*$ will make on the realizable subset of data. Therefore, this SOA should be executed with a mistake bit of 1 for every round in its history. In other words, executing every expert with mistake bit of 1 guarantees the existence of one with "small" number of mistakes.

Moreover, since information in the feedback is partial we cannot apply the classic tool from the setting of prediction with expert advice. Hence, we borrow tools from the bandits literature.

We describe the bandit setup we use in our proof. The game plays between a learner $\mathcal{A}$ and an adversary in $T$ rounds. We have a set $\{1,\ldots,K\}$ of $K \in \mathbb{N}$ actions and a set $\mathcal{E} = \{E_1,\ldots,E_N\}$ of $N \in \mathbb{N}$ experts. At round $t$, the adversary chooses an array of rewards $g(t) = (g^{(1)}(t),\ldots,g^{(K)}(t)) \in [0,1]^K$ for the actions but does not reveal it to the learner. Each expert $E_i$ (potentially based on the history) outputs a probability distribution $p_i(t) = (p_i^{(1)},\ldots,p_i^{(K)}) \in [0,1]^K$ with $\sum_{j\in[K]} p_i^{(j)} = 1$ over the set of actions. The learner $\mathcal{A}$ at each round $t$ observes the distributions $p_1(t),\ldots,p_N(t)$ of the experts based on which it computes a probability distribution $p_{\mathcal{A}}(t) = (p_{\mathcal{A}}^{(1)},\ldots,p_{\mathcal{A}}^{(K)}) \in [0,1]^K$ with $\sum_{j\in[K]} p_{\mathcal{A}}^{(j)} = 1$ and draws an action $\hat{y} \sim p_{\mathcal{A}}(t)$. The adversary then only reveals the reward $g^{(\hat{y})}(t)$. Define by $E_U$ the uniform (randomized) expert such that its probability distribution $p_U(t)$ satisfies $p_U^{(j)}(t) = 1/|K|$ for all $t \in T$ and $j \in K$. Auer et al. [ACFS02] introduce the EXP4 algorithm for the setting of non-stochastic bandits with prediction advice with guarantees summarized in the following lemma.

**Lemma 4.3** (Theorem 7.1 in [ACFS02])**.** *Let $K, N \in \mathbb{N}, \gamma \in (0,1]$. Let $\{1,\ldots,K\}$ be a set of actions and $\mathcal{E} = \{E_1,\ldots,E_N\}$ a set of $N$ experts with $E_U \in \mathcal{E}$. The algorithm* EXP4 *in [ACFS02]) satisfies*

$$\max_{i\in[K]} \sum_{i=1}^{T} \langle p_i(t), g(t)\rangle - \sum_{i=1}^{T} \langle p_{\mathrm{EXP4}}(t), g(t)\rangle \le (e-1)\gamma \max_{i\in[K]} \sum_{i=1}^{T} \langle p_i(t), g(t)\rangle + \frac{K\ln(N)}{\gamma}.$$

**Proof of Theorem 4.2.** Let $X = (x_1,\ldots,x_T)$, $Y = (y_1,\ldots,y_T)$, and $\mathcal{S} = (S_1,\ldots,S_T)$ be the online sequence. For any $n \in \mathbb{N}$ denote by $\mathbf{1}_n$ the all-ones sequence of size $n$. Let $\mathsf{SOA}_{\mathsf{known}}$ be the Mistake Known SOA learner defined in Algorithm 2. We will define a set $\mathcal{E}$ of experts as follows.

Let $\mathcal{I} = \{I \subseteq [T] : |I| \le \mathsf{LDknown}(\mathcal{H})\}$. For each set of indices $I \in \mathcal{I}$, we define an expert $E_I$ that in round $t$ predicts as $E_I(X_{<t}, Y_{<t})(x_t) = \mathsf{SOA}_{\mathsf{known}}(X_{I_{<t}}, Y_{I_{<t}}, \mathbf{1}_{t-1})(x_t)$, where $I_{<t} = \{i \in I : i < t\}$ is the subset of indices in $I$ that are smaller than $t$. In other words, at round $t$, the expert $E_I$ only has in its memory the indices in $I$ that appeared so far. Let $\mathcal{E} = \{E_I : I \in \mathcal{I}\} \cup \{E_U\}$, where $E_U$ is the uniform expert that always outputs a uniform distribution over the label set $\mathcal{Y}$.

Denote $E_I^{(t)} := E_I(X_{<t}, Y_{<t})(x_t)$. We will show that there exists an expert $E_{I^*}$ such that

$$\sum_{t=1}^{T} \mathbb{1}[E_{I^*}^{(t)} \notin S_t] \le \mathsf{LDknown}(\mathcal{H}) + \min_{h\in\mathcal{H}} \sum_{t=1}^{T} \mathbb{1}[h(x_t) \ne S_t]. \tag{9}$$

We first show how this concludes the claimed regret upper bound. We claim this problem can be reduced a non-stochastic bandits with expert advice problem. The adversary at each time $t$, will (potentially randomly and based on the history) select a tuple $(x_t, y_t, S_t)$. Each expert in $\mathcal{E}$ makes a prediction and the learner makes a prediction $\hat{y}_t$ based on these recommendations. The adversary then reveals the reward of choosing $\hat{y}_t$, i.e., it feeds back $g^{\hat{y}}(t) := \mathbb{1}[\hat{y}_t \in S_t]$. Note that $S_t$ can be viewed as defining a reward vector $g(t) \in \{0,1\}^{|\mathcal{Y}|}$ with $g^{\hat{y}}(t) := \mathbb{1}[\hat{y} \in S_t]$. Observe crucially that the prediction of experts at time $t$ depend only on the past history $(X_{<t}, Y_{<t})$ and the current instance $x_t$: the experts are SOAs that are only fed back (a subset of) the past history, with mistake bits being equal to 1, and the current instance.

We know $\mathcal{E}$ is a set of $\sum_{i\leq \mathsf{LDknown}(\mathcal{H})}\binom{T}{i} \leq (eT/\mathsf{LDknown}(\mathcal{H}))^{\mathsf{LDknown}(\mathcal{H})}$ experts over the actions space $\mathcal{Y}$. The algorithm $\mathcal{A}$ then uses the EXP4 algorithm [ACFS02] on these experts. Note that if $T \leq \frac{|\mathcal{Y}|\ln(|\mathcal{E}|)}{e-1}$, we obviously have a mistake (and, thus, regret) bound, of $T \leq O(\sqrt{\mathsf{LDknown}(\mathcal{H})|\mathcal{Y}|T\log(T)})$. Otherwise, by setting $\gamma = \sqrt{\frac{|\mathcal{Y}|\ln(|\mathcal{E}|)}{(e-1)T}}$ in Lemma 4.3 we immediately get

$$\max_{E\in\mathcal{E}}\sum_{i=1}^{T}\langle E_I^{(t)}, g(t)\rangle - \sum_{i=1}^{T}\langle p_{\mathcal{A}}(t), g(t)\rangle \leq 2\sqrt{(e-1)|\mathcal{Y}|T\mathsf{LDknown}(\mathcal{H})\ln\left(\frac{eT}{\mathsf{LDknown}(\mathcal{H})}\right)}.$$

On the other hand note that

$$\begin{aligned}\max_{E\in\mathcal{E}}\sum_{i=1}^{T}\langle E_I^{(t)}, g(t)\rangle - \sum_{i=1}^{T}\langle p_{\mathcal{A}}(t), g(t)\rangle &= \max_{E\in\mathcal{E}}\sum_{i=1}^{T}\mathbb{1}[E_I^{(t)}\in S_t] - \mathbb{E}\left[\sum_{i=1}^{T}\mathbb{1}[p_{\mathcal{A}}(t)\in S_t]\right]\\ &= \mathbb{E}\left[\sum_{i=1}^{T}\mathbb{1}[p_{\mathcal{A}}(t)\notin S_t]\right] - \min_{E\in\mathcal{E}}\sum_{i=1}^{T}\mathbb{1}[E_I^{(t)}\notin S_t].\end{aligned}$$

Therefore, we get that

$$\mathbb{E}\left[\sum_{i=1}^{T}\mathbb{1}[p_{\mathcal{A}}(t)\notin S_t]\right] - \min_{h\in\mathcal{H}}\sum_{t=1}^{T}\mathbb{1}[h(x_t)\neq S_t] \leq \mathsf{LDknown}(\mathcal{H}) + 4\sqrt{\mathsf{LDknown}(\mathcal{H})|\mathcal{Y}|T\log(T)},$$

as desired.

We turn to proving the existence of the expert claimed in (9). Let $h^*\in\arg\min_{h\in\mathcal{H}}\sum_{t=1}^{T}\mathbb{1}[h(x_t)\neq S_t]$ and $R\subseteq[T]$ be the subset of indices on which $h^*$ is correct. Clearly, $|R| \geq T - \min_{h\in\mathcal{H}}\sum_{t=1}^{T}\mathbb{1}[h(x_t)\neq S_t]$. Consider the following thought experiment. For any $t\in[|R|]$ let $R(t)\in[T]$ denote the index at position $t$ in $R$. If $\mathsf{SOA}_{\mathsf{known}}(\emptyset,\emptyset,0)(x_{R(1)})\in S_{R(1)}$, then discard $x_{R(1)}, y_{R(1)}$ and move to $R_2$. Repeat the same process until we get to the first instance $i_1^*$ such that $\mathsf{SOA}_{\mathsf{known}}(\emptyset,\emptyset,0)(x_{R(i_1^*)})\notin S_{R(i_1^*)}$. We keep $x_{R(i_1^*)}, y_{R(i_1^*)}$ in history. We continue this process for $|R|$ many times and will keep or discard the instances based on the correctness of the prediction.

Formally, let $I^*(0)=\emptyset$, $n_0=0$. For $t\in[|R|]$, if $\mathsf{SOA}_{\mathsf{known}}((X_{I^*(t-1)},Y_{I^*(t-1)}),\mathbf{1}_{n_{t-1}})(x_{R(t)})\in S_{R(t)}$, discard $x_{R(t)}, y_{R(t)}$, let $n_t = n_{t-1}$ and $I^*(t) = I^*(t-1)$. Otherwise, let $n_t = n_{t-1}+1$ and $I^*(t) = I^*(t-1)\cup R(t)$. Repeat this process until $t=|R|$ and let $I^* = I(|R|)$. Observe that this SOA will actually make a mistake on any $i^*\in I^*$ and since $R$ is realizable by $h^*$ we must have $|I^*|\leq\mathsf{LDknown}(\mathcal{H})$. Therefore, $I^*\in\mathcal{I}$ and hence $E_{I^*}\in\mathcal{E}$. In other words, the SOA that makes a mistake on every index in $I^*$ and is correct on any instance in $R\setminus I^*$ is included in $\mathcal{E}$. For this expert we have $\sum_{i=1}^{T}\mathbb{1}[E_{I^*}^{(t)}\notin S_t] \leq |I^*| + |[T]\setminus R| \leq \mathsf{LDknown}(\mathcal{H}) + \min_{h\in\mathcal{H}}\sum_{t=1}^{T}\mathbb{1}[h(x_t)\neq S_t]$, as desired. □

### 4.3 Agnostic Online Learning Under Set-valued Feedback Model

In this section we focus on regret in the set-valued agnostic online setting. We first give the characteristics of a family of hypothesis classes with regret bound that is at most logarithmic in the size of the class. Our next result is a regret upper bound of $O(\sqrt{\mathsf{LDset}(\mathcal{H})T\log(T)})$ for any class $\mathcal{H}$, which compared to Theorem 4.2 has no dependence on $|\mathcal{Y}|$ and works even for infinite label spaces.

**A characterization of hypothesis classes with constant regret.** In Section 4.1, we showed that a simple multi-label hypothesis class admits constant regret in the set-valued setting. The following theorem extends this result by giving a necessary and sufficient condition for constant regret for any finite class. In particular, the sets of correct labels assigned by every pair of hypotheses must intersect at every instance. When this condition holds, the regret is at most logarithmic in the size of the class.

**Theorem 4.4.** *A finite multi-label hypothesis class $\mathcal{H}$ admits $O(1)$ regret in the set-valued setting if and only if*

$$\forall x\in\mathcal{X},\ \forall h,h'\in\mathcal{H}:\quad h(x)\cap h'(x)\neq\emptyset.$$

*When this condition holds, there exists an algorithm $\mathcal{A}$ such that $\mathrm{R}_{\mathcal{A}}^{\mathrm{set}}(T,\mathcal{H}) \leq \log_2|\mathcal{H}|$ for every $T\in\mathbb{N}$.*

To prove the upper bound, we use a randomized multiplicative-weight algorithm. Initially, each hypothesis $h \in \mathcal{H}$ has weight 1. At each round $t$, the algorithm independently samples two hypotheses with probabilities proportional to their weights, and predicts any label in the intersection of the sets they assign to $x_t$. The condition in Theorem 4.4 guarantees that this intersection is nonempty. After observing $S_t$, it halves the weights of hypotheses whose label sets at $x_t$ differ from $S_t$, and keeps the remaining weights unchanged. We show that the expected loss in each round is at most the decrease in the logarithm of the sum of weights, which gives the stated regret bound. For the converse, if two hypotheses assign disjoint sets to some instance, an adversary can force regret of order $\Omega(\sqrt{T})$. The complete proof of Theorem 4.4 can be found in Appendix **??**.

**Sub-linear regret bound independent of label set size.** In section 4.2 we showed a regret bound that is sub-linear but scales with $O(\sqrt{|\mathcal{Y}|})$. The following theorem highlights that in the set-valued setting the regret can be made independent of label set size. The proof of the following theorem appears in Appendix D.3 and follows techniques similar to those used in Raman et al. [RST24], which in turn build on earlier proof techniques from Hanneke et al. [HMR+23].

**Theorem 4.5.** *For any multi-label hypothesis class $\mathcal{H}$, and any $T \geq \mathsf{LDset}(\mathcal{H})$,*

$$\inf_{\mathcal{A}} \mathrm{R}^{\mathrm{set}}_{\mathcal{A}}(\mathcal{H}, T) \leq O\left(\sqrt{\mathsf{LDset}(\mathcal{H}) T \log(T)}\right).$$

## 5 Conclusion

We studied learning from multiple correct answers. We formalized this problem in the online setting and under three models of feedback: mistake-unknown, mistake-known, and set-valued. For each model we characterized the mistake bound in the realizable setup, and we derived bounds on the regret in the agnostic setup. Our results demonstrate fundamental differences between this problem and the single-label setting. We further show how our results yield novel sample complexity bounds for the corresponding problems in the batch setup.

Our goal in this work is to model recent emerging uses of automated prediction. The formal framework(s) and the results stated in this paper provides insight to theoretically understanding such tasks. Many relevant aspects, however, are left unaddressed. Examples include non-binary evaluation of outputs, implicitly used prior knowledge beyond the framework of hypothesis classes, the computational complexity of learning and more.

## Acknowledgements

Shai Ben-David was supported by NSERC through a Discovery grant and by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute. Alireza F. Pour and Farnam Mansouri were supported by Vector Institute Research Grant and Cheriton Graduate Scholarship.

## References

[ACFS02] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire "The non-stochastic multiarmed bandit problem". In: *SIAM journal on computing* 32.1 (2002), pp. 48–77.

[BKM95] Shai Ben-David, Eyal Kushilevitz, and Yishay Mansour "Online learning versus offline learning". In: *European Conference on Computational Learning Theory*. Springer. 1995, pp. 38–52.

[Bla22] Moise Blanchard "Universal online learning: An optimistically universal learning rule". In: *Conference on Learning Theory*. PMLR. 2022, pp. 1077–1125.

[BPS09] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz "Agnostic Online Learning." In: *COLT*. Vol. 3. 2009, p. 1.

[DSBS15] Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz "Multiclass learnability and the ERM principle." In: *J. Mach. Learn. Res.* 16.1 (2015), pp. 2377–2404.

[HKLM22] Max Hopkins, Daniel M Kane, Shachar Lovett, and Gaurav Mahajan "Realizable learning is all you need". In: *Conference on Learning Theory*. PMLR. 2022, pp. 3015–3069.

[HMR+23] Steve Hanneke, Shay Moran, Vinod Raman, Unique Subedi, and Ambuj Tewari "Multiclass online learning and uniform convergence". In: *The Thirty Sixth Annual Conference on Learning Theory*. PMLR. 2023, pp. 5682–5696.

[HMS23] Steve Hanneke, Shay Moran, and Jonathan Shafer "A trichotomy for transductive online learning". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 19502–19519.

[HRSS24] Steve Hanneke, Vinod Raman, Amirreza Shaeiri, and Unique Subedi "Multiclass transductive online learning". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 60403–60428.

[JLB+25] Nirmit Joshi, Gene Li, Siddharth Bhandari, Shiva Prasad Kasiviswanathan, Cong Ma, and Nathan Srebro "Learning to Answer from Correct Demonstrations". In: *arXiv preprint arXiv:2510.15464* (2025).

[Joa02] Thorsten Joachims "Optimizing search engines using clickthrough data". In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2002, pp. 133–142.

[Lit88] Nick Littlestone "Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm". In: *Machine learning* 2.4 (1988), pp. 285–318.

[Liu+09] Tie-Yan Liu et al. "Learning to rank for information retrieval". In: *Foundations and Trends® in Information Retrieval* 3.3 (2009), pp. 225–331.

[LMS+17] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky "Adversarial Learning for Neural Dialogue Generation". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Ed. by Martha Palmer, Rebecca Hwa, and Sebastian Riedel. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2157–2169.

[LSC+21] Gen Li, Laixi Shi, Yuxin Chen, Yuantao Gu, and Yuejie Chi "Breaking the sample complexity barrier to regret-optimal model-free reinforcement learning". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 17762–17776.

[LZ07] John Langford and Tong Zhang "The epoch-greedy algorithm for contextual multi-armed bandits". In: *Advances in neural information processing systems* 20.1 (2007), pp. 96–1.

[MSTY23] Shay Moran, Ohad Sharon, Iska Tsubari, and Sivan Yosebashvili "List online classification". In: *The Thirty Sixth Annual Conference on Learning Theory*. PMLR. 2023, pp. 1885–1913.

[OWJ+22] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. "Training language models to follow instructions with human feedback". In: *Advances in neural information processing systems* 35 (2022), pp. 27730–27744.

[RST15] Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari "Online learning via sequential complexities". In: *The Journal of Machine Learning Research* 16.1 (2015), pp. 155–186.

[RST24] Vinod Raman, Unique Subedi, and Ambuj Tewari "Online learning with set-valued feedback". In: *The Thirty Seventh Annual Conference on Learning Theory*. PMLR. 2024, pp. 4381–4412.

[RST25] Vinod Raman, Unique Subedi, and Ambuj Tewari "A Unified Theory of Supervised Online Learnability". In: *36th International Conference on Algorithmic Learning Theory*. 2025.

[SB14] Shai Shalev-Shwartz and Shai Ben-David *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

## A Learning from Multiple Correct Answers in the Batch Setup

In this section, we show our online results imply learnability in different PAC settings too. We use Online-to-Batch conversion techniques and show the relevant combinatorial parameters defined in Section 3 can control the sample complexity of learning from multiple answers in the batch setting. We start with the main lemma that we exploit to turn our online learners into batch learners, the proof of which can be found in Appendix E. We then give formal definition of different Batch settings we consider.

**Lemma A.1.** *Let $\mathcal{A} : (\mathcal{X} \times \mathcal{Z})^* \to \Pi(\mathcal{Y})^{\mathcal{X}}$ and $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Z}$. For any $U := ((x_1, z_1), ..., (x_m, z_m)) \in (\mathcal{X} \times \mathcal{Z})^*$ define $\mathcal{A}_{\text{o2b}} : (\mathcal{X} \times \mathcal{Z})^* \to \Pi(\mathcal{Y})^{\mathcal{X}}$ as*

$$\forall x \in \mathcal{X} : \mathcal{A}_{o2b}(U)(x) \sim \text{Unif}\left(f_1(x), \ldots, f_m(x)\right), \text{ where } \forall t \in [m] : f_t(x) := \mathcal{A}(U_{<t})(x).$$

*Then for any $\varepsilon, \delta \in (0, 1] \times (0, 1]$ and $h^* : \mathcal{X} \to 2^{\mathcal{Y}}$, with probability at least $1 - \delta$ over an i.i.d generated sample $U := ((x_1, z_1), \ldots, (x_m, z_m))$ from $\mathcal{D}$ we have*

$$\mathop{\mathbb{P}}_{x \sim \mathcal{D}_{\mathcal{X}}}[\mathcal{A}_{\text{o2b}}(U) \notin h^*(x)] \leq \frac{1 + \sum_{t=1}^{m} \ell_t + 12 \log\left(\frac{2 \log m}{\delta}\right)}{m},$$

*where $\ell_t := \mathbb{1}\left[f_t\left(x_t\right) \notin h^*\left(x_t\right)\right]$.*

We now give the definition of realizable multi-label learning. We say a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ is realizable by a function $h : \mathcal{X} \mapsto 2^{\mathcal{Y}}$, if $\mathbb{P}_{(x,y)\sim\mathcal{D}}[y \in h(x)] = 1$.

**Definition A.2** (Realizable Multi-Label Learning, Definition 1 of [JLB+25])**.** *Let $\mathcal{H}$ be a multi-label hypothesis class. We say that $\mathcal{H}$ is* realizable multi-label learnable *if there exist a function $m^{\text{unknown}} : (0, 1) \times (0, 1) \to \mathbb{N}$, and a learner $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^* \to \Pi(\mathcal{Y})^{\mathcal{X}}$ such that for any $h^* \in \mathcal{H}$, and distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ realizable by $h^*$, any $(\varepsilon, \delta) \in (0, 1) \times (0, 1)$, if $m \geq m^{\text{unknown}}(\varepsilon, \delta)$, with probability at least $1 - \delta$ over i.i.d. generated sample $U = ((x_1, y_1), \ldots, (x_m, y_m))$ from $\mathcal{D}$ we have*

$$\mathop{\mathbb{P}}_{x \sim \mathcal{D}_{\mathcal{X}}}[\mathcal{A}(U)(x) \notin h^*(x)] < \varepsilon,$$

*where $\mathcal{D}_{\mathcal{X}}$ is the marginal distribution of $\mathcal{D}$.*

The following corollary exhibits that the sample complexity of PAC learning with multiple labels in the realizable setting is controlled by LDunknown. The proof simply follows by combining Lemma E.1 and the mistake upper bound of $\mathsf{LDunknown}(\mathcal{H})$ in Theorem 3.7. Noting that for any finite class $\mathcal{H}$, we have $\mathsf{LDunknown}(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$ (see Corollary 3.8), the following generalizes Theorem 5 of Joshi et al. [JLB+25] to infinite hypothesis classes, resolving one of their open questions.

**Corollary A.3.** *Every multi-label hypothesis class with bounded* LDunknown *is realizable multi-label learnable with sample complexity*

$$m^{\text{unknown}}(\varepsilon, \delta) = O\left(\varepsilon^{-1}\left(\mathsf{LDunknown}(\mathcal{H}) + \log(1/\varepsilon\delta)\right)\right)$$

Next, we focus on set-valued PAC learnability. Here, we need a set-valued variant of Lemma A.1, which we state below. Since the proof is identical to that of Lemma A.1, it is not repeated here.

**Lemma A.4.** *Let $\mathcal{A} : \left(\mathcal{X} \times 2^{\mathcal{Y}}\right)^* \to \Pi(\mathcal{Y})^{\mathcal{X}}$ and $\mathcal{D}$ be a distribution over $\mathcal{X} \times 2^{\mathcal{Y}}$. For any $U := ((x_1, S_1), \ldots, (x_m, S_m)) \in (\mathcal{X} \times 2^{\mathcal{Y}})^*$ define $\mathcal{A}_{\text{o2b}} : \left(\mathcal{X} \times 2^{\mathcal{Y}}\right)^* \to \Pi(\mathcal{Y})^{\mathcal{X}}$ as*

$$\forall x \in \mathcal{X} : \mathcal{A}_{o2b}(U)(x) \sim \text{Unif}\left(f_1(x), \ldots, f_m(x)\right), \text{ where } \forall t \in [m] : f_t(x) := \mathcal{A}(U_{<t})(x).$$

*Then for any $\varepsilon, \delta \in (0, 1] \times (0, 1]$, with probability at least $1 - \delta$ over an i.i.d generated sample $U := ((x_1, S_1), \ldots, (x_m, S_m))$ from $\mathcal{D}$ we have*

$$\mathop{\mathbb{P}}_{(x,S)\sim\mathcal{D}}[\mathcal{A}_{\text{o2b}}(U) \notin S] \leq \frac{1 + \sum_{t=1}^{m} \ell_t + 12 \log\left(\frac{2 \log m}{\delta}\right)}{m},$$

*where $\ell_t := \mathbb{1}\left[f_t\left(x_t\right) \notin S_t\right]$.*

We first give the definition set-valued learning in the realizable setup. Then similar to realizable multi-label learning, in Corollary A.6, we show that the sample complexity of realizable set-Valued learning is controlled by LDset. Corollary A.6 immediately follows from Lemma A.4 and Theorem 3.7.

**Definition A.5** (Realizable Set-Valued Learning)**.** *Let $\mathcal{H}$ be a multi-label hypothesis class. We say that $\mathcal{H}$ is* realizable set-valued learnable *if there exist function $m^{\text{set}} : (0,1) \times (0,1) \to \mathbb{N}$, and a learner $\mathcal{A} : \left(\mathcal{X} \times 2^{\mathcal{Y}}\right)^* \to \Pi(\mathcal{Y})^{\mathcal{X}}$ such that for any $h^* \in \mathcal{H}$, any distribution $\mathcal{D}_{\mathcal{X}}$ over $\mathcal{X}$, any $(\varepsilon, \delta) \in (0,1) \times (0,1)$, if $m \geq m^{\text{set}}(\varepsilon, \delta)$, with probability at least $1 - \delta$ over an i.i.d. generated sample $U = ((x_1, h^*(x_1)), \ldots, (x_m, h^*(x_m)))$ from $\mathcal{D}$ we have*

$$\mathbb{P}_{x \sim \mathcal{D}_{\mathcal{X}}}[\mathcal{A}(U)(x) \notin h^*(x)] < \varepsilon.$$

**Corollary A.6.** *Every multi-label hypothesis class with bounded* LDset *is realizable set-valued learnable with sample complexity*

$$m^{\text{set}}(\varepsilon, \delta) = O\left(\varepsilon^{-1}\left(\mathsf{LDset}(\mathcal{H}) + \log(1/\varepsilon\delta)\right)\right)$$

Finally, we define set-valued learning in the agnostic setup and provide sample complexity upper bound for this problem in Theorem A.8. The proof of Theorem A.8 combines Lemma A.4, Theorem **??**, and Theorem 4.5. The formal proof can be found in Appendix E.

**Definition A.7** (Agnostic Set-valued Learning)**.** *Let $\mathcal{H}$ be a multi-label hypothesis class. We say that $\mathcal{H}$ is* agnostic set-valued learnable *if there exist function $m^{\text{set}} : (0,1) \times (0,1) \to \mathbb{N}$, and a learner $\mathcal{A} : \left(\mathcal{X} \times 2^{\mathcal{Y}}\right)^* \to \Pi(\mathcal{Y})^{\mathcal{X}}$ such that for any distribution $\mathcal{D}$ over $\mathcal{X} \times 2^{\mathcal{Y}}$, any $(\varepsilon, \delta) \in (0,1) \times (0,1)$, if $m \geq m^{\text{set}}(\varepsilon, \delta)$, with probability at least $1 - \delta$ over an i.i.d. generated sample $U = ((x_1, S_1), \ldots, (x_m, S_m))$ from $\mathcal{D}$ we have*

$$\mathbb{P}_{(x,S) \sim \mathcal{D}}[\mathcal{A}(U)(x) \notin S] < \inf_{h \in \mathcal{H}} \mathbb{P}_{(x,S) \sim \mathcal{D}}[h(x) \neq S] + \varepsilon.$$

**Theorem A.8.** *Every multi-label hypothesis class with bounded* LDset *is realizable set-valued learnable with sample complexity*

$$O\left(\frac{\mathsf{LDset}(\mathcal{H})}{\varepsilon^2} \log\left(\frac{\mathsf{LDset}(\mathcal{H})}{\varepsilon^2}\right) + \frac{\log(1/\delta)}{\varepsilon^2}\right).$$

*Moreover, for any finite-sized multi-label hypothesis class $\mathcal{H}$ such that*

$$\forall h, h' \in \mathcal{H} \text{ and } \forall x \in \mathcal{X}: \quad 2|h'(x) \setminus h(x)| \leq |h(x)|,$$

*is agnostic set-valued learnable with sample complexity*

$$O\left(\varepsilon^{-1}\left(\log|\mathcal{H}|\right) + \varepsilon^{-2}\log(1/\delta)\right).$$

## B Related Works

Recently, Joshi et al. [JLB+25] and Raman et al. [RST24] studied learning with multiple correct answers under the mistake-unknown and set-valued feedback models, respectively. Both works focus primarily on the online learning setting, with Joshi et al. [JLB+25] additionally considering implications for batch learning.

In contrast to our work, Joshi et al. [JLB+25] restrict attention to finite hypothesis classes, whereas we allow for possibly infinite classes. For the realizable setting, their framework is similar to our mistake-unknown feedback model. However, they only derive an upper bound for finite classes in this case, whereas we provide a full (upper and lower bounds) characterization. In the agnostic setting, however, the loss measures used in Joshi et al. [JLB+25] differ from ours, and as a result our guarantees are not directly comparable. We discuss these differences and the advantages of our setup in detail in Section 2.1.

Turning to Raman et al. [RST24], their notion of regret is again different from ours (see Section 2.1). Furthermore, we also show that in the agnostic set-valued setting the regret of a variety of hypothesis classes can be constant. There is no similar result in previous work.

At a higher level perspective, beyond multi-label classification, our results are situated within the analysis of online learnability and its combinatorial characterizations, originating from binary

**Algorithm 1** Mistake Unknown Multi-label Standard Optimal Algorithm (SOA)

**Input:** Multi-label hypothesis class $\mathcal{H}$ and $T \in \mathbb{N}$.

Initialize $V_1 = \mathcal{H}$, $\mathsf{off}_1 \equiv 0$.
**for** $t = 1$ **to** $T$ **do**
  Receive $x_t$
  For $r \in \mathcal{Y}$, define $V_t^{(r)} := \{h \in V_t : r \in h(x_t)\}$
  For every $\hat{r} \in \mathcal{Y}$ and $h \in \mathcal{H}$ define $\mathsf{off}_t^{(\hat{r})}(h) := \mathsf{off}_t(h) + \mathbb{1}[\hat{r} \notin h(x_t)]$
  Predict $\hat{y}_t := \arg\min_{\hat{r} \in \mathcal{Y}} \max_{r \in \mathcal{Y}} \mathsf{LDunknown}(V_t^{(r)}, \mathsf{off}_t^{(\hat{r})})$
  Receive label $y_t$
  Set $V_{t+1} = V_t^{(y_t)}$, and $\mathsf{off}_{t+1} = \mathsf{off}_t^{(\hat{y}_t)}$.

classification and the Littlestone dimension [Lit88; BPS09], and extending to multi-class learning [DSBS15; HMR+23], transductive learning [BKM95; HMS23; HRSS24], and frameworks for online learning with general (local) loss functions [RST15; Bla22; RST25]. Moran et al. [MSTY23] study an opposite prediction paradigm, in which there is a single correct label per instance, but the learner is allowed to output a list of labels, where the loss is defined by whether the correct label is contained in the list.

# C Missing Proofs from Section 3

## C.1 Proof of Theorem 3.5: Characterization of Classes with Dimension Zero

We first establish a simple structural lemma in the mistake-unknown setup, which we use to prove the theorem.

**Lemma C.1.** *If* $\mathsf{LDunknown}(\mathcal{H}) \geq d$*, then there exists a multi-label tree* $\mathcal{T}$ *that is* $d$*-shattered by* $\mathcal{H}$*, such that for every path* $\sigma = (\hat{y}_1, ..., \hat{y}_T)$ *from root to leaf there exists a* $h_\sigma$ *that in addition to satisfying properties (i) and (ii) also satisfies* $\hat{y}_T \notin h_\sigma(\nu_x(\sigma_{<T}))$.

*Proof.* Consider any tree $\mathcal{T}$ that is $d$-shattered by $\mathcal{H}$. Suppose there exists a root to leaf path $\sigma = (\hat{y}_1, ..., \hat{y}_T)$ with $\hat{y}_T \in h_\sigma(\nu_x(\sigma_{<T}))$. Note that the function $h_\sigma$ still satisfies properties (i) and (ii) for the path $\sigma' = (\hat{y}_1, ..., \hat{y}_{T-1})$. This implies that if we remove all the children of the node $\nu(\sigma_{\leq T-1})$, the reduced tree will still be $d$-shattered by $\mathcal{H}$. □

**Theorem 3.5.** *For any multi-label hypothesis class* $\mathcal{H}$*, we have*

$$\mathsf{LDset}(\mathcal{H}) = 0 \;\Leftrightarrow\; \mathsf{LDknown}(\mathcal{H}) = 0 \;\Leftrightarrow\; \mathsf{LDunknown}(\mathcal{H}) = 0 \;\Leftrightarrow\; \forall x \in \mathcal{X} : \bigcap_{h \in \mathcal{H}} h(x) \neq \emptyset.$$

*Proof.* Since $\mathsf{LDset}(\mathcal{H}) \leq \mathsf{LDknown}(\mathcal{H}) \leq \mathsf{LDunknown}(\mathcal{H})$ it suffice to show that $\forall x \in \mathcal{X} : \bigcap_{h \in \mathcal{H}} h(x) \neq \emptyset$ implies $\mathsf{LDunknown}(\mathcal{H}) = 0$ and $\mathsf{LDset}(\mathcal{H}) = 0$ implies $\forall x \in \mathcal{X} : \bigcap_{h \in \mathcal{H}} h(x) \neq \emptyset$.

$\forall x \in \mathcal{X} : \bigcap_{h \in \mathcal{H}} h(x) \neq \emptyset \implies \mathsf{LDunknown}(\mathcal{H}) = 0$: for the sake of contradiction suppose that $\mathsf{LDunknown}(\mathcal{H}) \geq 1$. Let $\mathcal{T}$ be a tree promised in Lemma C.1 such that $\mathcal{T}$ is $\mathsf{LDunknown}(\mathcal{H})$-shattered by $\mathcal{H}$. Consider a node of tree such that all its children are a leaf. Denote $x$ as the node's $\mathcal{X}-$value. Due to Lemma C.1, for every $\hat{r} \in \mathcal{Y}$, there exists a $h_{\hat{r}} \in \mathcal{H}$ such that $\hat{r} \notin h_{\hat{r}}(x)$. Thus $\bigcap_{h \in \mathcal{H}} h(x) = \emptyset$, which is a contradiction.

$\mathsf{LDset}(\mathcal{H}) = 0 \implies \forall x \in \mathcal{X} : \bigcap_{h \in \mathcal{H}} h(x) \neq \emptyset$: for the sake of contradiction suppose that there exists $x \in \mathcal{X}$ such that for every $\hat{r} \in \mathcal{Y}$ there exists a $h_{\hat{r}} \in \mathcal{H}$ such that $\hat{r} \notin h_{\hat{r}}(x)$. Let $\mathcal{T}$ be a set-valued tree of depth 1 with root $x$ and $\forall \hat{r} \in \mathcal{Y} : \nu_S(\hat{r}) = h_{\hat{r}}(x)$. Then $\mathcal{T}$ is shattered by $\mathcal{H}$, which implies $\mathsf{LDset}(\mathcal{H}) \geq 1$ which is a contradiction. □

**Algorithm 2** Mistake Known Multi-label Standard Optimal Algorithm (SOA)

**Input:** Multi-label hypothesis class $\mathcal{H}$ and $T \in \mathbb{N}$.

Initialize $V_1 = \mathcal{H}$.
**for** $t = 1$ **to** $T$ **do**
  Receive $x_t$
  For $r, \hat{r} \in \mathcal{Y}$, define $V_t^{(r,\hat{r},1)} := \{h \in V_t : r \in h(x_t), \hat{r} \notin h(x_t)\}$ and $V_t^{(r,\hat{r},0)} := \{h \in V_t : r \in h(x_t), \hat{r} \in h(x_t)\}$
  **if** $\mathsf{LDknown}(V_t) > 0$ **then**
    Predict $\hat{y}_t := \arg\min_{\hat{r} \in \mathcal{Y}} \max_{r \in \mathcal{Y}} \mathsf{LDknown}(V_t^{(r,\hat{r},1)})$
  **else**
    Predict any $\hat{y}_t \in \bigcap_{h \in V_t} h(x_t)$ (exists due to Theorem 3.5)
  Receive label $y_t$ and $b_t$ indicating whether $\hat{y}_t$ was mistake
  Set $V_{t+1} = V_t^{(y_t, \hat{y}_t, b_t)}$

**Algorithm 3** Set-valued Standard Optimal Algorithm (SOA)

**Input:** Multi-label hypothesis class $\mathcal{H}$ and $T \in \mathbb{N}$.

Initialize $V_1 = \mathcal{H}$.
**for** $t = 1$ **to** $T$ **do**
  Receive $x_t$
  For $S \subseteq \mathcal{Y}$ and $\hat{r} \in \mathcal{Y}$ define $V_t^{(S,\hat{r})} := \{h \in V_t : h(x_t) = S, \hat{r} \notin h(x_t)\}$
  **if** $\mathsf{LDset}(V_t) > 0$ **then**
    Predict $\hat{y}_t := \arg\min_{\hat{r} \in \mathcal{Y}} \max_{S \subseteq \mathcal{Y}} \mathsf{LDset}(V_t^{(S,\hat{r})})$
  **else**
    Predict any $\hat{y}_t \in \bigcap_{h \in V_t} h(x_t)$ (exists due to Theorem 3.5)
  Receive set feedback $S_t$ indicating whether $\hat{y}_t$ was mistake
  Set $V_{t+1} = V_t^{(S_t, \hat{y}_t)}$

### C.2 Proof of Theorem 3.7: Realizable Characterizations

**Theorem 3.7.** *For any multi-label hypothesis class $\mathcal{H}$, and any $T \in \mathbb{N}$ we have the following characterizations of the optimal mistake bound for the three feedback models:*

$$\begin{aligned}
&\textit{Mistake-unknown feedback:} && \inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}^{\mathrm{unknown}}(T, \mathcal{H}) = \min(\mathsf{LDunknown}(\mathcal{H}), T).\\
&\textit{Mistake-known feedback:} && \inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}^{\mathrm{known}}(T, \mathcal{H}) = \min(\mathsf{LDknown}(\mathcal{H}), T).\\
&\textit{Set-valued feedback:} && \inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}^{\mathrm{set}}(T, \mathcal{H}) = \min(\mathsf{LDset}(\mathcal{H}), T).
\end{aligned}$$

*Proof.* We prove the characterization for each feedback model in the following.

**Mistake-unknown characterization:** We first prove that for any $T \geq \mathsf{LDunknown}(\mathcal{H})$ and any algorithm $\mathcal{A}$ we have $\mathrm{M}_{\mathcal{A}}^{\mathrm{unknown}}(T, \mathcal{H}) \geq \mathsf{LDunknown}(\mathcal{H})$. Let $\mathcal{T}$ be a multi-label tree that is $\mathsf{LDunknown}(\mathcal{H})$-shattered by $\mathcal{H}$ and let $x_1$ denote its root. For every $t \in [\mathsf{LDunknown}(\mathcal{H})]$, denote $\hat{y}_t := p_{\mathcal{A}}(t)$, $y_t = \nu_y(\hat{y}_1, \ldots, \hat{y}_t)$, and $x_{t+1} = \nu_x(\hat{y}_1, \ldots, \hat{y}_t)$. Let $h^\star = h_{\sigma^\star}$ for $\sigma^\star = (\hat{y}_1, \ldots, \hat{y}_{\mathsf{LDknown}(\mathcal{H})})$. From property (i) of Definition 3.3 we have $y_t \in h^\star(x_t)$. Moreover, due to property (ii) $\sum_{t=1}^{\mathsf{LDunknown}(\mathcal{H})} \mathbb{1}[\hat{y}_t \notin h^\star(x_t)] \geq \mathsf{LDunknown}(\mathcal{H})$. This concludes that $\mathcal{A}$ makes at least $\mathsf{LDunknown}(\mathcal{H})$ mistakes and the claim follows.

Next we show that for the learner $\mathcal{A}$ defined in Algorithm 1 and any $T \in \mathbb{N}$ we have $\mathrm{M}_{\mathcal{A}}^{\mathrm{unknown}}(T, \mathcal{H}) \leq \mathsf{LDunknown}(\mathcal{H})$. We prove this by showing that $\mathsf{LDunknown}(V_t, \mathsf{off}_t)$ is non-

increasing in $t$. Note that this finishes the proof since for $h^\star \in \mathcal{H}$,

$$\begin{aligned}
\mathsf{LDunknown}(\mathcal{H}) &= \mathsf{LDunknown}(V_1, \mathsf{off}_1) \\
&\geq \mathsf{LDunknown}(V_{T+1}, \mathsf{off}_{T+1}) && \text{(since } \mathsf{LDunknown}(V_t, \mathsf{off}_t) \text{ is non-increasing)} \\
&\geq \sup_{h \in V_{T+1}} \mathsf{off}_{T+1}(h) && \text{(due to Claim 3.4)} \\
&\geq \mathsf{off}_{T+1}(h^\star) \\
&= \mathrm{M}^{\mathrm{unknown}}_{\mathcal{A}}(T, \mathcal{H}),
\end{aligned}$$

where the last equality follows from the fact that we can inductively prove

$$\mathsf{off}_{T+1}(h^\star) = \mathsf{off}_T(h^\star) + \mathbb{1}[\hat{y}_T \notin h^\star(x_T)] = \sum_{t=1}^{T} \mathbb{1}[\hat{y}_t \notin h^\star(x_t)] = \sum_{t=1}^{T} \mathbb{1}[p_{\mathcal{A}}(t) \notin h^\star(x_t)].$$

We now prove that $\mathsf{LDunknown}(V_t, \mathsf{off}_t)$ is non-increasing in $t$. For the sake of contradiction suppose $\mathsf{LDunknown}(V_{t+1}, \mathsf{off}_{t+1}) > \mathsf{LDunknown}(V_t, \mathsf{off}_t)$ for some $t \in [T]$. For every $\hat{r} \in \mathcal{Y}$ denote

$$y_{\mathsf{max}}(\hat{r}) := \arg\max_r \mathsf{LDunknown}(V_t^{(r)}, \mathsf{off}_t^{(\hat{r})}).$$

Note that from the description of Algorithm 1, we have

$$\hat{y}_t = \arg\min_{\hat{r} \in \mathcal{Y}} \mathsf{LDunknown}(V_t^{(y_{\mathsf{max}}(\hat{r}))}, \mathsf{off}_t^{(\hat{r})}).$$

Therefore, for all $\hat{r} \in \mathcal{Y}$, we have,

$$\begin{aligned}
\mathsf{LDunknown}(V_t^{(y_{\mathsf{max}}(\hat{r}))}, \mathsf{off}_t^{(\hat{r})}) &\geq \mathsf{LDunknown}(V_t^{(y_{\mathsf{max}}(\hat{y}_t))}, \mathsf{off}_t^{(\hat{y}_t)}) \\
&\geq \mathsf{LDunknown}(V_t^{(y_t)}, \mathsf{off}_t^{(\hat{y}_t)}) \\
&= \mathsf{LDunknown}(V_{t+1}, \mathsf{off}_{t+1}).
\end{aligned}$$

For any $\hat{r} \in \mathcal{Y}$, we let $\mathcal{T}_{\hat{r}}$ be a multi-label tree with $\mathsf{LDunknown}(V_t^{(y_{\mathsf{max}}(\hat{r}))}, \mathsf{off}_t^{(\hat{r})}) \geq \mathsf{LDunknown}(V_{t+1}, \mathsf{off}_{t+1})$ that is shattered by $\left(V_t^{(y_{\mathsf{max}}(\hat{r}))}, \mathsf{off}_t^{(\hat{r})}\right)$, where the label on the root is $y_{\mathsf{max}}(\hat{r})$. Construct the multi-label tree $\overline{\mathcal{T}}$ such that root of $\overline{\mathcal{T}}$ is $x_t$, and each edge $\hat{r} \in \mathcal{Y}$ of the root is connected to $\mathcal{T}_{\hat{r}}$. Then $\overline{\mathcal{T}}$ is shattered by $(V_t, \mathsf{off}_t)$. This indicates that $\mathsf{LDunknown}(V_t, \mathsf{off}_t) \geq \mathsf{LDunknown}(V_{t+1}, \mathsf{off}_{t+1})$ which is a contradiction.

**Mistake-known characterization:** We first show that for any $T \geq \mathsf{LDknown}(\mathcal{H})$ and any algorithm $\mathcal{A}$ we have $\mathrm{M}^{\mathrm{known}}_{\mathcal{A}}(T, \mathcal{H}) \geq \mathsf{LDknown}(\mathcal{H})$. Find a complete multi-label tree $\mathcal{T}$ of depth $\mathsf{LDknown}(\mathcal{H})$ that is shattered by $\mathcal{H}$. Let $x_1$ be the root of $\mathcal{T}$. For every $t \in [\mathsf{LDknown}(\mathcal{H})]$, define $\hat{y}_t := p_{\mathcal{A}}(t)$, $y_t = \nu_y(\hat{y}_1, ..., \hat{y}_t)$ and $x_{t+1} = \nu_x(\hat{y}_1, ..., \hat{y}_t)$. Let $h^\star = h_{\sigma^\star}$ for $\sigma^\star = (\hat{y}_1, \ldots, \hat{y}_{\mathsf{LDknown}(\mathcal{H})})$. First note that due to property (i) of Definition 3.2 we have $y_t \in h^\star(x_t)$. Moreover due to property (ii), $\hat{y}_t \notin h^\star(x_t)$. This concludes $\mathrm{M}^{\mathrm{known}}_{\mathcal{A}}(T, \mathcal{H}) \geq \mathsf{LDknown}(\mathcal{H})$.

Next we prove that for the learner $\mathcal{A}$ defined in Algorithm 2 and any $T \in \mathbb{N}$, we have $\mathrm{M}^{\mathrm{known}}_{\mathcal{A}}(T, \mathcal{H}) \leq \mathsf{LDknown}(\mathcal{H})$. We prove this by showing for each $t$, if $\mathsf{LDknown}(V_t) > 0$ and $\mathcal{A}$ makes a mistake, then $\mathsf{LDknown}(V_{t+1}) \leq \mathsf{LDknown}(V_t) - 1$. Note that this concludes the claim since Theorem 3.5 ensures that whenever $\mathsf{LDknown}(V_t) = 0$, $\bigcap_{h \in V_t} h(x_t) \neq \emptyset$ and $\mathcal{A}$ makes no more mistakes.

For the sake of contradiction suppose $\mathcal{A}$ makes a mistake at $t$ and $\mathsf{LDknown}(V_{t+1}) = \mathsf{LDknown}(V_t)$. For every $\hat{r} \in \mathcal{Y}$ denote $y_{\mathsf{max}}(\hat{r}) := \arg\max_r \mathsf{LDknown}(V_t^{(r,\hat{r},1)})$. Then, noting that $m_t = 1$, we have

$$\begin{aligned}
\mathsf{LDknown}(V_t^{(y_{\mathsf{max}}(\hat{r}),\hat{r},1)}) &\geq \mathsf{LDknown}(V_t^{(y_{\mathsf{max}}(\hat{y}_t),\hat{y}_t,1)}) \geq \mathsf{LDknown}(V_t^{(y_t,\hat{y}_t,1)}) \\
&= \mathsf{LDknown}(V_{t+1}) = \mathsf{LDknown}(V_t).
\end{aligned}$$

For any $\hat{r} \in \mathcal{Y}$, let $\mathcal{T}_{\hat{r}}$ be a complete multi-label tree of depth $\mathsf{LDknown}(V_t)$ that is shattered by $V_t^{(y_{\mathsf{max}}(\hat{r}),\hat{r},1)}$, where the label on the root is $y_{\mathsf{max}}(\hat{r})$. Construct the complete multi-label tree $\overline{\mathcal{T}}$ such that root of $\overline{\mathcal{T}}$ is $x_t$, and each edge $\hat{r} \in \mathcal{Y}$ of the root is connected to $\mathcal{T}_{\hat{r}}$. Clearly $\overline{\mathcal{T}}$ is

shattered by $\bigcup_{\hat{r}} V_t^{(y_{\max}(\hat{r}),\hat{r},1)}$, which implies that it is also shattered by $V_t \supseteq \bigcup_{\hat{r}} V_t^{(y_{\max}(\hat{r}),\hat{r},1)}$. Hence, $\mathsf{LDknown}(V_t) \geq \mathsf{LDknown}(V_t) + 1$ which is a contradiction.

**Set-valued characterization:** We first show that for any $T \geq \mathsf{LDset}(\mathcal{H})$ and any algorithm $\mathcal{A}$ we have $\mathrm{M}^{\mathrm{set}}_{\mathcal{A}}(T, \mathcal{H}) \geq \mathsf{LDset}(\mathcal{H})$. Find a set-valued tree $\mathcal{T}$ of depth $\mathsf{LDset}(\mathcal{H})$ that is shattered by $\mathcal{H}$. Let $x_1$ be the root of $\mathcal{T}$. For every $t \in [\mathsf{LDset}(\mathcal{H})]$, define $\hat{y}_t := p_{\mathcal{A}}(t)$, $S_t = \nu_S(\hat{y}_1, ..., \hat{y}_t)$ and $x_{t+1} = \nu_x(\hat{y}_1, ..., \hat{y}_t)$. Let $h^\star = h_{\sigma^\star}$ for $\sigma^\star = (\hat{y}_1, \ldots, \hat{y}_{\mathsf{LDset}(\mathcal{H})})$. First note that due to property (i) of Definition 3.1 we have $S_t = h^\star(x_t)$. Moreover due to property (ii), $\hat{y}_t \notin h^\star(x_t)$. This concludes $\mathrm{M}^{\mathrm{set}}_{\mathcal{A}}(T, \mathcal{H}) \geq \mathsf{LDset}(\mathcal{H})$.

Next we prove that for the learner $\mathcal{A}$ defined in Algorithm 3 and any $T \in \mathbb{N}$, we have $\mathrm{M}^{\mathrm{set}}_{\mathcal{A}}(T, \mathcal{H}) \leq \mathsf{LDset}(\mathcal{H})$. We prove this by showing at each $t$, if $\mathsf{LDset}(V_t) > 0$ and $\mathcal{A}$ makes a mistake, then $\mathsf{LDset}(V_{t+1}) \leq \mathsf{LDset}(V_t) - 1$. Note that this concludes the claim since Theorem 3.5 ensures that whenever $\mathsf{LDset}(V_t) = 0$, we have $\bigcap_{h \in V_t} h(x_t) \neq \emptyset$ and $\mathcal{A}$ makes no more mistakes.

For the sake of contradiction suppose $\mathcal{A}$ makes a mistake at $t$ and $\mathsf{LDset}(V_{t+1}) = \mathsf{LDset}(V_t)$. For every $\hat{r} \in \mathcal{Y}$ denote $S_{\mathsf{max}}(\hat{r}) := \arg\max_{S \subseteq \mathcal{Y}} \mathsf{LDset}(V_t^{(S,\hat{r})})$

$$\mathsf{LDset}(V_t^{(S_{\mathsf{max}}(\hat{r}),\hat{r})}) \geq \mathsf{LDset}(V_t^{(S_{\mathsf{max}}(\hat{y}_t),\hat{y}_t)}) \geq \mathsf{LDset}(V_t^{(S_t,\hat{y}_t)}) = \mathsf{LDset}(V_{t+1}) = \mathsf{LDset}(V_t).$$

Let $\mathcal{T}_{\hat{r}}$ be a complete multi-label tree of depth $\mathsf{LDset}(V_t)$ that is shattered by $V_t^{(S_{\mathsf{max}}(\hat{r}),\hat{r})}$, where the set of labels on the root is $S_{\mathsf{max}}(\hat{r})$. Construct the complete multi-label tree $\overline{\mathcal{T}}$ such that root of $\overline{\mathcal{T}}$ is $x_t$, and each edge $\hat{r} \in \mathcal{Y}$ of the root is connected to $\mathcal{T}_{\hat{r}}$. Clearly $\overline{\mathcal{T}}$ is shattered by $\bigcup_{\hat{r}} V_t^{(S_{\mathsf{max}},\hat{r})}$, which implies that it is also shattered by $V_t \supseteq \bigcup_{\hat{r}} V_t^{(S_{\mathsf{max}}(\hat{r}),\hat{r})}$. Hence, $\mathsf{LDset}(V_t) \geq \mathsf{LDset}(V_t) + 1$ which is a contradiction. □

### C.3 Proof of Theorem 3.6: Relationships Between the Dimensions

**Theorem 3.6.** *For every $n \geq 1$, there exists a multi-label hypothesis class $\mathcal{H}_n$ over a finite label space such that*

$$\mathsf{LDset}(\mathcal{H}_n) = 1, \qquad \mathsf{LDknown}(\mathcal{H}_n) = n, \qquad \mathsf{LDunknown}(\mathcal{H}_n) = \infty.$$

*Moreover, for $\mathsf{LD} \in \{\mathsf{LDset}, \mathsf{LDknown}\}$ and any $\mathcal{H}_1, \mathcal{H}_2$, $\mathsf{LD}(\mathcal{H}_1 \cup \mathcal{H}_2) \leq \mathsf{LD}(\mathcal{H}_1) + \mathsf{LD}(\mathcal{H}_2) + 1$. This inequality does not hold for $\mathsf{LDunknown}$, even when $\mathsf{LDunknown}(\mathcal{H}_1) = \mathsf{LDunknown}(\mathcal{H}_2) = 0$.*

*Proof.* Fix $n \geq 1$. Let $\mathcal{X} = \mathbb{N}$ and $\mathcal{Y} = [n+1] \cup \{a, b\}$, where $a, b$ are two additional labels. For each $i \in [n+1]$, let $\mathcal{G}_i$ be the class of all functions satisfying

$$h(x) \in \big\{\{i, a\}, \{i, b\}\big\} \qquad \text{for every } x \in \mathcal{X},$$

and set $\mathcal{H}_n = \bigcup_{i=1}^{n+1} \mathcal{G}_i$. Thus, the value of $i$ is fixed within each $\mathcal{G}_i$, while the choice between $a$ and $b$ can be made independently at every instance.

**Set-valued dimension.** The learner can predict $a$ in the first round. Upon observing $S_1$, it learns the unique $i \in [n+1]$ contained in $S_1$, and can predict $i$ in every subsequent round. Hence $\mathsf{LDset}(\mathcal{H}_n) \leq 1$. The reverse inequality follows from Theorem 3.5, since $\bigcap_{h \in \mathcal{H}_n} h(x) = \emptyset$ for every $x \in \mathcal{X}$.

**Mistake-known dimension.** For the upper bound, the learner keeps the set $I \subseteq [n+1]$ of indices consistent with the feedback. It predicts any $i \in I$. If this is a mistake, then $i$ is removed from $I$, since only hypotheses in $\mathcal{G}_i$ include the label $i$. Otherwise, the learner can continue to predict $i$ without further mistakes. Thus, there are at most $n$ mistakes.

For the lower bound, let the adversary present a fresh instance in each round and maintain a set $I$ of possible indices, initially $[n+1]$. If the learner predicts $a$ (respectively, $b$), the adversary returns the correct label $b$ (respectively, $a$), and all indices in $I$ remain possible. If the learner predicts $i \in [n+1]$, the adversary returns $a$ as the correct label and indicates a mistake, eliminating $i$ from $I$ if $i \in I$. In the latter case, the adversary can choose a consistent label set $\{j, a\}$ for any $j \in I \setminus \{i\}$. Since at most one index is eliminated in each round, for the first $n$ rounds there remains a hypothesis consistent

**Algorithm 4** Set-Valued Weighted Majority Algorithm

**Input:** Multi-label hypothesis class $\mathcal{H}$ and $T \in \mathbb{N}$.

Initialize $w^{(1)}(h) = 1$ for all $h \in \mathcal{H}$.
**for** $t = 1$ **to** $T$ **do**
  Receive $x_t$.
  For every $y \in \mathcal{Y}$ define $v^{(t)}(y) := \sum_{h \in \mathcal{H} \text{ s.t. } y \in h(x_t)} w^{(t)}(h)$.
  Output $\hat{y}_t = y$ with probability $\frac{v^{(t)}(y)}{\sum_{y \in \mathcal{Y}} v^{(t)}(y)}$.
  Receive $S_t$.
  **if** $\hat{y}_t \notin S_t$ **then**
    For each $h \in \mathcal{H}$, update

$$w^{(t+1)}(h) \leftarrow \begin{cases} 2w^{(t)}(h) & \text{if } h(x_t) = S_t, \\ w^{(t)}(h) & \text{otherwise.} \end{cases}$$

  **else**
    For each $h \in \mathcal{H}$, update

$$w^{(t+1)}(h) \leftarrow \begin{cases} w^{(t)}(h) & \text{if } h(x_t) = S_t, \\ w^{(t)}(h)/2 & \text{otherwise.} \end{cases}$$

with all the feedback, and the learner makes a mistake in every round. It follows from Theorem 3.7 that $\mathsf{LDknown}(\mathcal{H}_n) = n$.

**Mistake-unknown dimension.** Fix $T \in \mathbb{N}$. Consider a multi-label tree of depth $T$ in which every node at level $t$ has instance $x_t = t$. For each branch corresponding to a prediction $\hat{y}_t$, let the observed label be $y_t = b$ if $\hat{y}_t = a$, and $y_t = a$ otherwise. Along any root-to-leaf path, choose an index $i \in [n+1]$ that appears least often among the predictions. Since the instances are distinct, there is a hypothesis $h \in \mathcal{G}_i$ with $h(x_t) = \{i, y_t\}$ for all $t \in [T]$. The observed labels are consistent with $h$, and the number of mistakes of the predictions against $h$ is

$$\sum_{t=1}^{T} \mathbb{1}[\hat{y}_t \notin h(x_t)] = T - \sum_{t=1}^{T} \mathbb{1}[\hat{y}_t = i] \geq T - \frac{T}{n+1}.$$

Hence this tree is $\lfloor nT/(n+1) \rfloor$-shattered by $\mathcal{H}_n$. As $T$ is arbitrary, $\mathsf{LDunknown}(\mathcal{H}_n) = \infty$.

**Unions.** Let $\mathsf{LD}$ be either $\mathsf{LDset}$ or $\mathsf{LDknown}$, and denote $d_i = \mathsf{LD}(\mathcal{H}_i)$ for $i = 1, 2$. It suffices to consider finite $d_1, d_2$. Run the corresponding SOA for $\mathcal{H}_1$ as long as its feedback is consistent with a hypothesis in $\mathcal{H}_1$. If the feedback becomes inconsistent, or if the algorithm has made $d_1 + 1$ mistakes, switch to the SOA for $\mathcal{H}_2$. A realizable sequence generated by a hypothesis in $\mathcal{H}_1$ never causes a switch, and the SOA makes at most $d_1$ mistakes. Otherwise, once $d_1 + 1$ mistakes have occurred, realizability by $\mathcal{H}_1$ is ruled out. The learner therefore makes at most $d_1 + 1$ mistakes before the switch and at most $d_2$ afterward. Theorem 3.7 gives

$$\mathsf{LD}(\mathcal{H}_1 \cup \mathcal{H}_2) \leq d_1 + d_2 + 1.$$

Finally, in the construction above take $\mathcal{G}_1$ and $\mathcal{G}_2$ for $n = 1$. Each class has mistake-unknown dimension zero, since its hypotheses share the label $i$ at every instance. However, their union has infinite mistake-unknown dimension, as shown above. This proves the last statement. □

# D Missing Proofs from Section 4

## D.1 Proof of Theorem 4.1: Regret for Example Hypothesis Class

**Theorem 4.1.** *Consider the hypothesis class $\mathcal{H}_3 = \{h_1, h_2, h_3\}$ over domain $\mathcal{X} = \{x\}$ and label set $\mathcal{Y} = \{1, 2, 3\}$, where $h_1(x) = \{2, 3\}$, $h_2(x) = \{1, 3\}$, and $h_3(x) = \{1, 2\}$. We have $\mathsf{LDunknown}(\mathcal{H}_3) = \mathsf{LDset}(\mathcal{H}_3) = 1$, whereas*

$$\inf_{\mathcal{A}} \mathrm{R}_{\mathcal{A}}^{\mathrm{unknown}}(T, \mathcal{H}_3) \geq \frac{T}{9}, \quad \inf_{\mathcal{A}} \mathrm{R}_{\mathcal{A}}^{\mathrm{set}}(T, \mathcal{H}_3) \leq \log_2(3).$$

*Proof.* Since there is only a single input $x$, for the sake of brevity we write $h$ instead of $h(x)$ for every $h \in \mathcal{H}_3$.

(i) Fix a learner $\mathcal{A}$. For each $t \in [T]$, let $y_t$ be sampled uniformly from $\mathcal{Y}$. For each $y \in \mathcal{Y}$, define $L_y := \sum_{t=1}^{T} \Pr[p_{\mathcal{A}}(t) = y]$. Furthermore, let $\hat{y} := \arg\max_{y \in \mathcal{Y}} L_y$. Since $\sum_{y \in \mathcal{Y}} L_y = T$, it follows that $L_{\hat{y}} \geq \frac{T}{3}$. Now, for each $t \in [T]$, define

$$S_t := \begin{cases} \{\hat{y}\} & \text{if } y_t = \hat{y}, \\ h_{\hat{y}} & \text{o.w.} \end{cases}$$

Then we have

$$\begin{aligned} \sum_{t=1}^{T} \Pr[p_{\mathcal{A}}(t) \notin S_t] &= \sum_{t=1}^{T} \Big( \Pr[p_{\mathcal{A}}(t) = \hat{y}] \Pr[y_t \neq \hat{y}] + \Pr[p_{\mathcal{A}}(t) \neq \hat{y}] \Pr[y_t = \hat{y}] \Big) \\ &= \sum_{t=1}^{T} \left( \frac{2}{3} \Pr[p_{\mathcal{A}}(t) = \hat{y}] + \frac{1}{3} \Pr[p_{\mathcal{A}}(t) \neq \hat{y}] \right) \\ &= \frac{T}{3} + \frac{1}{3} \sum_{t=1}^{T} \Pr[p_{\mathcal{A}}(t) = \hat{y}] \\ &= \frac{T}{3} + \frac{L_{\hat{y}}}{3} \;\geq\; \frac{4T}{9}. \end{aligned} \tag{10}$$

On the other hand,

$$\begin{aligned} \mathbb{E}\left[ \inf_{h \in \mathcal{H}_3} \sum_{t=1}^{T} 1\{h \neq S_t\} \right] &\leq \mathbb{E}\left[ \sum_{t=1}^{T} 1\{h_{\hat{y}} \neq S_t\} \right] \\ &= \sum_{t=1}^{T} \Pr[y_t = \hat{y}] = \frac{T}{3}. \end{aligned}$$

Combining this with (10) completes the proof.

(ii) Let $\mathcal{A}$ be the algorithm defined in Algorithm 4. For every $t \in [T]$, denote

$$W^{(t)} := \sum_{h \in \mathcal{H}_3} w^{(t)}(h).$$

We first show that $\mathbb{E}\left[W^{(t)}\right]$ is always non-increasing with respect to $t$. Notice that if $S_t \notin \mathcal{H}$, either the learner makes a mistake in which case the weights remain unchanged or the learner does not make a mistake in which case all weights will be cut in half. In any case $W^{(t+1)} \leq W^{(t)}$. Otherwise, without loss of generality, suppose $S_t = \{1, 2\}$. Notice that

$$\Pr[\hat{y}_t \notin \{1,2\}] = \frac{w^{(t)}(h_1) + w^{(t)}(h_2)}{2W^{(t)}}, \quad \Pr[\hat{y}_t \in \{1,2\}] = \frac{w^{(t)}(h_1) + w^{(t)}(h_2) + 2w^{(t)}(h_3)}{2W^{(t)}}$$

Therefore,

$$\begin{aligned} \mathbb{E}\left[ W^{(t+1)} \mid w^{(t)} \right] &= W^{(t)} + w^{(t)}(h_3) \cdot \frac{w^{(t)}(h_1) + w^{(t)}(h_2)}{2W^{(t)}} \\ &\quad - \frac{w^{(t)}(h_1) + w^{(t)}(h_2)}{2} \cdot \frac{w^{(t)}(h_1) + w^{(t)}(h_2) + 2w^{(t)}(h_3)}{2W^{(t)}} \\ &= W^{(t)} - \frac{\left(w^{(t)}(h_1) + w^{(t)}(h_2)\right)^2}{4W^{(t)}} < W^{(t)}. \end{aligned}$$

This proves the claim. Next, for every $h \in \mathcal{H}_3$, define

$$R(h) := \sum_{t=1}^{T} \mathbb{1}\left[p_{\mathcal{A}}(t) \notin S_t\right] - \mathbb{1}\left[h \neq S_t\right].$$

Observe that by the definition of $w^{(t+1)}$, for every $h \in \mathcal{H}_3$,

$$\mathbb{1}\left[p_{\mathcal{A}}(t) \notin S_t\right] - \mathbb{1}\left[h \neq S_t\right] = \log_2\left(\frac{w^{(t+1)}(h)}{w^{(t)}(h)}\right).$$

Therefore, $R(h) = \log_2\big(w^{(T+1)}(h)\big)$. Let $h^* := \arg\max_{h \in \mathcal{H}_3} w^{(T+1)}(h)$. By Jensen's inequality, we have

$$2^{R^{\mathrm{set}}_{\mathcal{A}}(T,\mathcal{H}_3)} = 2^{\mathbb{E}[R(h^*)]} \leq \mathbb{E}\Big[2^{R(h^*)}\Big] = \mathbb{E}\Big[w^{(T+1)}(h^*)\Big] \leq \mathbb{E}\Big[W^{(T+1)}\Big] \leq W^{(1)} = |\mathcal{H}_3|.$$

This completes the proof.

□

### D.2 Proof of Theorem 4.4: Constant Set-Valued Regret

**Theorem 4.4.** *A finite multi-label hypothesis class $\mathcal{H}$ admits $O(1)$ regret in the set-valued setting if and only if*

$$\forall x \in \mathcal{X},\ \forall h, h' \in \mathcal{H}: \quad h(x) \cap h'(x) \neq \emptyset.$$

*When this condition holds, there exists an algorithm $\mathcal{A}$ such that $\mathrm{R}^{\mathrm{set}}_{\mathcal{A}}(T, \mathcal{H}) \leq \log_2 |\mathcal{H}|$ for every $T \in \mathbb{N}$.*

*Proof.* We first prove necessity. Suppose there exist $x \in \mathcal{X}$ and $h_0, h_1 \in \mathcal{H}$ such that $h_0(x) \cap h_1(x) = \emptyset$. If either set is empty, the adversary can choose $S_t = \emptyset$ and $x_t = x$ in every round, forcing regret $T$. Otherwise, consider an adversary that presents $x$ in every round and chooses $S_t = h_0(x)$ or $S_t = h_1(x)$ independently with equal probability. Since the two sets are disjoint, every learner has expected cumulative loss at least $T/2$. On the other hand, if $N$ is the number of rounds on which $S_t = h_0(x)$, then

$$\min_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{1}[h(x) \neq S_t] \leq \min\{N, T - N\} = \frac{T}{2} - \left|N - \frac{T}{2}\right|.$$

As $N \sim \mathrm{Bin}(T, 1/2)$, the expected regret is at least $\mathbb{E}[|N - T/2|] = \Omega(\sqrt{T})$. Hence no algorithm has constant regret.

For sufficiency, assume that $h(x) \cap h'(x) \neq \emptyset$ for every $x \in \mathcal{X}$ and $h, h' \in \mathcal{H}$. Initialize $w_1(h) = 1$ for every $h \in \mathcal{H}$. On round $t$, after observing $x_t$, define

$$q_t(h) = \frac{w_t(h)}{\sum_{h' \in \mathcal{H}} w_t(h')}.$$

Independently draw $h_t, h'_t \sim q_t$ and predict any $\hat{y}_t \in h_t(x_t) \cap h'_t(x_t)$. The assumption guarantees that such a label exists. After observing $S_t$, update

$$w_{t+1}(h) = w_t(h)\, 2^{-\mathbb{1}[h(x_t) \neq S_t]}.$$

Let $W_t = \sum_{h \in \mathcal{H}} w_t(h)$ and $p_t = \sum_{h : h(x_t) = S_t} q_t(h)$. If either sampled hypothesis equals $S_t$ at $x_t$, then the prediction belongs to $S_t$. Consequently,

$$\Pr[\hat{y}_t \notin S_t \mid x_t, S_t, w_t] \leq (1 - p_t)^2, \qquad \frac{W_{t+1}}{W_t} = \frac{1 + p_t}{2}.$$

Using the inequality $(1 - p)^2 \leq -\log_2((1 + p)/2)$ for every $p \in [0, 1]$, we obtain

$$\Pr[\hat{y}_t \notin S_t \mid x_t, S_t, w_t] \leq \log_2 \frac{W_t}{W_{t+1}}.$$

Summing over $t$ and taking expectations gives

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}[\hat{y}_t \notin S_t]\right] \leq \mathbb{E}\left[\log_2 \frac{W_1}{W_{T+1}}\right].$$

For every $h \in \mathcal{H}$ we have $w_{T+1}(h) = 2^{-\sum_{t=1}^{T} \mathbb{1}[h(x_t) \neq S_t]}$. Therefore,

$$W_{T+1} \geq 2^{-\min_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{1}[h(x_t) \neq S_t]}.$$

Since $W_1 = |\mathcal{H}|$, it follows that

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}[\hat{y}_t \notin S_t]\right] \leq \min_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{1}[h(x_t) \neq S_t] + \log_2 |\mathcal{H}|,$$

which completes the proof. □

### D.3 Proof of Theorem 4.5: Set-Valued Regret Upper Bound

In contrast to the mistake-unknown setting, in the set-valued setting we have full information in the feedback. In other words, the learner observes the entire set of labels $S_t$ after prediction and can determine for every expert whether they made a mistake (as opposed to the mistake-known where the mistake bit is only known for the experts that recommended the predicted label). Hence, we can leverage the tools from the prediction with expert advice literature to get tighter bounds.

Let $T, N \in \mathbb{N}$ and $\mathcal{E} = \{E_1, \ldots, E_N\}$ be a set of $N$ experts such that each expert $E_i$ predicts a single label at each time $t$, which we denote by $E_i^{(t)}$. For $t \in [T], i \in [N]$ denote by $b_i(t) := \mathbb{1}[E_i^{(t)} \notin S_t]$ the bit that indicates whether $E_i$ made a mistake at time $t$. Let MWU be the algorithm that starts by setting a weight of $w_i(1) = 1$ to each expert. At each round $t \geq 2$ the learner updates $w_i(t) = w_i(t-1)e^{-\eta b_{i-1}(t)}$ for some $\eta > 0$. Let $Z_t = \sum_{i \in [N]} w_i(t)$. MWU then randomly draws an expert $E_i$ with probability $w_i(t)/Z_t$ and predicts according the the selected expert.

**Lemma D.1** (Theorem 21.11 of Shalev-Shwartz and Ben-David [SB14])**.** *Let $\eta = \sqrt{2\log(N)/T}$ and assume $T > 2\log(N)$. For the MWU algorithm defined above we have*

$$\sum_{i=1}^{T} \mathbb{1}[p_{MWU}(t) \notin S_t] - \min_{i \in [N]} \sum_{i=1}^{T} \mathbb{1}[E_I(t) \notin S_t] \leq \sqrt{2\log(N)T}.$$

**Theorem 4.5.** *For any multi-label hypothesis class $\mathcal{H}$, and any $T \geq \mathsf{LDset}(\mathcal{H})$,*

$$\inf_{\mathcal{A}} \mathrm{R}^{\mathrm{set}}_{\mathcal{A}}(\mathcal{H}, T) \leq O\left(\sqrt{\mathsf{LDset}(\mathcal{H})T\log(T)}\right).$$

*Proof.* For any subset $I \subseteq [T]$ of size at most $\mathsf{LDset}(\mathcal{H})$ we will create an expert $E_I$ that predicts according to the prediction of the Set-valued SOA that has in its history only the past information revealed in rounds in $I$. Let $\mathcal{I} = \{I \subseteq [T] : |I| \leq \mathsf{LDset}(\mathcal{H})\}$. We then define $\mathcal{E} = \{E_I : I \in \mathcal{I}\}$ as the set of all such experts which satisfies $|\mathcal{E}| \leq (eT/\mathsf{LDset}(\mathcal{H}))^{\mathsf{LDset}(\mathcal{H})}$. Note that similar to the arguments in the proof of Theorem 4.2, we can prove that each expert defined above is a valid expert since the SOAs just make their prediction based on the past history of $X_{<t}, Y_{<t}, S_{<t}$. Moreover, similar to proof of Thoerem 4.2, and from the guarantee that set-valued SOA will make at most $\mathsf{LDset}(\mathcal{H})$ mistakes on the longest realizable subsequence of $[T]$, we can see that there exists an expert $E_{I^*}$ in the set $\mathcal{E}$ defined above such that

$$\sum_{t=1}^{T} \mathbb{1}[E_{i^*}^{(t)} \notin S_t] \leq \mathsf{LDset}(\mathcal{H}) + \min_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{1}[h(x_t) \neq S_t].$$

Now, for the learner $\mathcal{A}$ that applies MWU on these sets of experts we have from Lemma D.1 that

$$\sum_{i=1}^{T} \mathbb{1}[p_{\mathcal{A}}(t) \notin S_t] - \min_{E_I \in \mathcal{E}} \sum_{i=1}^{T} \mathbb{1}[E_I(t) \notin S_t] \leq \sqrt{2\mathsf{LDset}(\mathcal{H})T\ln\left(\frac{eT}{\mathsf{LDset}(\mathcal{H})}\right)}.$$

This implies that

$$\sum_{i=1}^{T} \mathbb{1}[p_{\mathcal{A}}(t) \notin S_t] - \min_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{1}[h(x_t) \neq S_t] \leq \mathsf{LDset}(\mathcal{H}) + \sqrt{2\mathsf{LDset}(\mathcal{H})T\ln(T)}$$

and concludes the proof. □

## E Missing Proofs from Section A: Online-to-Batch Conversion

**Notations.** For any $h^* \in \mathcal{H}$ and $f \in \mathcal{Y}^{\mathcal{X}}$ and distribution $\mathcal{D}$ over $\mathcal{X}$ define $\mathsf{err}_{\mathcal{D},h^*}(f) = \Pr_{x\sim\mathcal{D}_{\mathcal{X}}}[f(x) \notin h^*(x)]$. Moreover, for any distribution $\mathcal{D}$ over $\mathcal{X} \times 2^{\mathcal{Y}}$, define $\mathsf{err}_{\mathcal{D}}(f) = \Pr_{(x,S)\sim\mathcal{D}}[f(x) \notin S]$. For any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Z}$, we denote by $\mathcal{D}_{\mathcal{X}}$ the marginal distribution of $\mathcal{D}$ over $\mathcal{X}$. Throughout Section A, we instantiate this result with $\mathcal{Z}$ as $\mathcal{Y}$ (Corollary A.3) or $2^{\mathcal{Y}}$ (Corollary A.6 and Theorem A.8) based on the learning model we consider.

**Lemma E.1** (Freedman's inequality, Theorem 3 of [LSC+21]). *Consider a filtration $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \mathcal{F}_2 \subset \ldots$. Let $Y_m = \sum_{i=1}^m X_i$, where $(X_i)_{i\in[m]}$ is a real-valued scalar sequence such that for some constant $R < \infty$, $|X_i| \le R$ and $\mathbb{E}[X_i \mid \mathcal{F}_{i-1}] = 0$. Define the predictable variance process*

$$W_m := \sum_{i=1}^{m} \mathbb{E}\left[X_i^2 \mid \mathcal{F}_{i-1}\right],$$

*and assume deterministically that $W_m \le \sigma^2$ for some constant $\sigma^2 < \infty$. Then for any integer $n \ge 1$, with probability at least $1-\delta$,*

$$|Y_m| \le \sqrt{8\max\left\{W_m, \frac{\sigma^2}{2^n}\right\}\log\left(\frac{2n}{\delta}\right)} + \frac{4}{3}R\log\left(\frac{2n}{\delta}\right).$$

### E.1 Proof of Lemma A.1

Due to definition of $f_t$, we have

$$\mathbb{E}[\ell_t \mid U_{<t}] = \mathsf{err}_{\mathcal{D}_{\mathcal{X}},h^*}(f_t).$$

Hence

$$\mathbb{E}[\mathsf{err}_{\mathcal{D}_{\mathcal{X}},h^*}(\mathcal{A}_{\mathrm{o2b}}(U))] = \mathbb{E}\left[\frac{1}{m}\sum_{t=1}^{m}\mathsf{err}_{\mathcal{D}_{\mathcal{X}},h^*}(f_t)\right] = \mathbb{E}\left[\frac{1}{m}\sum_{t=1}^{m}\ell_t\right].$$

For the high-probability statement, define the martingale differences

$$M_t := \mathsf{err}_{\mathcal{D}}(f_t) - \ell_t, \qquad \text{where } |M_t| \le 1 \text{ almost surely.}$$

Then $\mathbb{E}[M_t \mid U_{<t}] = 0$, and

$$\begin{aligned}\mathbb{E}\left[M_t^2 \mid U_{<t}\right] &= \mathbb{E}\left[\left(\mathsf{err}_{\mathcal{D}_{\mathcal{X}},h^*}(f_t) - \ell_t\right)^2 \mid U_{<t}\right] = \mathrm{Var}\left(\ell_t \mid U_{<t}\right)\\ &= \mathsf{err}_{\mathcal{D}_{\mathcal{X}},h^*}(f_t)\left(1 - \mathsf{err}_{\mathcal{D}_{\mathcal{X}},h^*}(f_t)\right) \le \mathsf{err}_{\mathcal{D}_{\mathcal{X}},h^*}(f_t).\end{aligned}$$

And taking $W_m = \sum_{t=1}^m \mathsf{err}_{\mathcal{D}_{\mathcal{X}},h^*}(f_t)$ and $\sigma^2 = m$ suffices, thus, using Lemma E.1 with $n = \log m$ inequality gives us with probability $1-\delta$

$$\begin{aligned}\sum_{t=1}^{m} M_t &\le \sqrt{8\left(1 + \sum_{t=1}^{m}\mathsf{err}_{\mathcal{D}_{\mathcal{X}},h^*}(f_t)\right)\log\left(\frac{2\log m}{\delta}\right)} + \frac{4}{3}\log\left(\frac{2\log m}{\delta}\right)\\ &\le \frac{1}{2}\left(1 + \sum_{t=1}^{m}\mathsf{err}_{\mathcal{D}_{\mathcal{X}},h^*}(f_t)\right) + 4\log\left(\frac{2\log m}{\delta}\right) + \frac{4}{3}\log\left(\frac{2\log m}{\delta}\right).\end{aligned}$$

Substituting $M_t$ and rearranging terms,

$$\sum_{t=1}^{m}\mathsf{err}_{\mathcal{D}_{\mathcal{X}},h^*}(f_t) \le 1 + 2\sum_{t=1}^{m}\ell_t + \frac{32}{3}\log\left(\frac{2\log m}{\delta}\right)$$

Finally noting that $\mathsf{err}_{\mathcal{D}_{\mathcal{X}},h^*}(\mathcal{A}_{\mathrm{o2b}}(U)) = \frac{1}{m}\sum_{t=1}^m \mathsf{err}_{\mathcal{D}_{\mathcal{X}},h^*}(f_t)$ completes the proof. □

### E.2 Proof of Theorem A.8

We only prove the first part. The second part would be similar to the first part. Let $h^* \in \arg\min_{h\in\mathcal{H}} \mathbb{P}_{(x,S)\sim\mathcal{D}}[S \neq h(x)]$. By a Chernoff's bound we know that if $m \geq 2\log(4/\delta)/\varepsilon^2$ then with probability $1-\delta/2$ over $U \sim \mathcal{D}^m$ we have

$$\frac{1}{m}\sum_{(x,S)\in U} \mathbb{1}[h^*(x) \neq S] \leq \mathop{\mathbb{P}}_{(x,S)\sim\mathcal{D}}[h^*(x) \neq S] + \frac{\varepsilon}{2}. \tag{11}$$

From Theorem 4.5 we know that for any set $U = ((x_1, S_1)\ldots,(x_m, S_m))$ of size $m \geq \mathsf{LDset}(\mathcal{H})$, there exists an algorithm $\mathcal{A}$ such that

$$\frac{1}{m}\sum_{t=1}^{m} \mathbb{1}[\mathcal{A}(U_{<t})(x) \notin S_t] - \min_{h\in\mathcal{H}} \frac{1}{m}\sum_{t=1}^{m} \mathbb{1}[h(x_t) \neq S_t] \leq \frac{\mathsf{LDset}(\mathcal{H})}{m} + \frac{\sqrt{2\mathsf{LDset}(\mathcal{H})m\ln(m)}}{m}.$$

We now invoke Lemma A.1 and take a union bound with Equation (11) to conclude that there exists a learner $\mathcal{A}_{o2b}$ such that with probability at least $1-\delta$ over $U \sim \mathcal{D}^m$,

$$\begin{aligned}
&\mathop{\mathbb{P}}_{(x,S)\sim\mathcal{D}}[\mathcal{A}_{\mathrm{o2b}}(U) \notin S] \leq \frac{1+\sum_{t=1}^{m}\mathbb{1}[\mathcal{A}(U_{<t})(x)\notin S_t]}{m} + 12\frac{\log\left(\frac{2\log m}{\delta}\right)}{m} \\
&\leq \frac{1}{m} + \frac{\min_{h\in\mathcal{H}}\sum_{t=1}^{m}\mathbb{1}[h(x_t)\neq S_t] + \mathsf{LDset}(\mathcal{H}) + \sqrt{2\mathsf{LDset}(\mathcal{H})m\ln(m)}}{m} + 12\frac{\log\left(\frac{2\log m}{\delta}\right)}{m} \\
&\leq \frac{1}{m} + \frac{\sum_{t=1}^{m}\mathbb{1}[h^*(x_t)\neq S_t] + \mathsf{LDset}(\mathcal{H}) + \sqrt{2\mathsf{LDset}(\mathcal{H})m\ln(m)}}{m} + 12\frac{\log\left(\frac{2\log m}{\delta}\right)}{m} \\
&\leq \frac{1}{m} + \mathop{\mathbb{P}}_{(x,S)\sim\mathcal{D}}[h^*(x)\neq S] + \frac{\varepsilon}{2} + \frac{\mathsf{LDset}(\mathcal{H}) + \sqrt{2\mathsf{LDset}(\mathcal{H})m\ln(m)}}{m} + 12\frac{\log\left(\frac{2\log m}{\delta}\right)}{m}.
\end{aligned}$$

Letting $m \geq O\left(\frac{\mathsf{LDset}(\mathcal{H})}{\varepsilon^2}\log\left(\frac{\mathsf{LDset}(\mathcal{H})}{\varepsilon^2}\right) + \log(1/\epsilon\delta)\right)$ completes the proof.

## NeurIPS Paper Checklist

1. **Claims**
   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   Answer: [Yes]
   Justification: We make claims about mistake bounds and regret, which are then (in the main body) formalized in a mathematical model and proven formally (either in the main paper or in the appendix).
   Guidelines:
   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.
2. **Limitations**
   Question: Does the paper discuss the limitations of the work performed by the authors?
   Answer: [Yes]
   Justification: Our conclusions section contains a subsection on limitations, which discusses limitations of this paper.
   Guidelines:
   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.
3. **Theory assumptions and proofs**
   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?
   Answer: [Yes]
   Justification: All assumptions are explicitly stated in the formulation of the formal results themselves. All formal proofs are provided either in the main paper or in the appendix.
   Guidelines:
   - The answer [N/A] means that the paper does not include theoretical results.

- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**
   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?
   Answer: [N/A]
   Justification: The paper does not contain experiments.
   Guidelines:
   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**
   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?
   Answer: [N/A]
   Justification: The paper does not contain experiments.
   Guidelines:
   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not

including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**
   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?
   Answer: [N/A]
   Justification: The paper does not contain experiments.
   Guidelines:
   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.
7. **Experiment statistical significance**
   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?
   Answer: [N/A]
   Justification: The paper does not contain experiments.
   Guidelines:
   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.
8. **Experiments compute resources**
   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?
   Answer: [N/A]
   Justification: The paper does not contain experiments.
   Guidelines:
   - The answer [N/A] means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed. The work described in this paper is purely theoretical and not tied to any particular applications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This is a purely theoretical paper that provides no data or models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring

that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**
Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?
Answer: [N/A]
Justification: The paper does not use existing assets.
Guidelines:
- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**
Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?
Answer: [N/A]
Justification: The paper does not release new assets.
Guidelines:
- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**
Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?
Answer: [N/A]
Justification: The paper does not involve crowdsourcing nor research with human subjects.
Guidelines:
- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.