Conformal Prediction Sets with Limited False Positives

Adam Fisch [1] **Tal Schuster** [2] **Tommi Jaakkola** [1] **Regina Barzilay** [1]

Abstract

We develop a new approach to multi-label conformal prediction in which we aim to output a precise set of promising prediction candidates with a bounded number of incorrect answers. Standard conformal prediction provides the ability to adapt to model uncertainty by constructing a calibrated candidate set in place of a single prediction, with guarantees that the set contains the correct answer with high probability. In order to obey this coverage property, however, conformal sets can become inundated with noisy candidates—which can render them unhelpful in practice. This is particularly relevant to practical applications where there is a limited budget, and the cost (monetary or otherwise) associated with false positives is non-negligible. We propose to trade coverage for a notion of precision by enforcing that the presence of incorrect candidates in the predicted conformal sets (i.e., the total number of false positives) is bounded according to a user-specified tolerance. Subject to this constraint, our algorithm then optimizes for a generalized notion of set coverage (i.e., the true positive rate) that allows for any number of true answers for a given query (including zero). We demonstrate the effectiveness of this approach across a number of classification tasks in natural language processing, computer vision, and computational chemistry.

1 Introduction

For many classification problems, returning a set of plausible responses instead of a single prediction is a useful way of representing uncertainty (Gammerman and Vovk, 2007; Lei, 2014; Romano et al., 2020). Conformal prediction (Vovk et al., 2005) is an increasingly popular method for creating confident prediction sets that provably contain the correct answer with high probability. Unfortunately, these guarantees do not come for free; in order to achieve proper coverage on difficult tasks, conformal prediction can often be unable to rule out an overwhelming number of candidates—making their prediction sets large and inefficient. This can make conformal predictors unusable in settings in which the cost of returning false positive predictions is substantial.

As an example, consider in-silico screening for drug discovery (see Figure 1). In-silico screening uses computational tools to search over millions of molecular compounds to identify candidates with desired properties. Any identified candidates are then verified experimentally. While it is often not necessary to return *all* possible viable candidates (e.g., even identifying just one effective drug can suffice), it is important to respect budgetary constraints by avoiding false positive predictions. Too many false positives can quickly consume available resources (e.g., time, materials, funding, or other assets). This is especially relevant when a valid answer, in this case, an effective drug, might not even exist.

In this work, we develop an approach to creating confident prediction sets that trades off standard coverage guarantees for practical, provable constraints on the total *number of false positives* (FP). In other words, we shift the focus of our conformal guarantees to be on limiting the number of incorrect answers in our outputs, with the understanding that we can potentially fail to recover some proportion of the true answers—i.e., we may obtain a lower *true positive rate* (TPR), which we assume is acceptable for the application.

Concretely, we are interested in a set prediction setting where we have been given n multi-label classification examples $(X_i, Z_i) \in \mathcal{X} \times 2^{\mathcal{Y}}$, $i = 1, \dots n$ as calibration data, that have been drawn exchangeably from some underlying distribution P_{XZ}. Under our assumptions, each observation X_i can be associated with any number of correct labels (including zero, in the case of having no answer at all, or one, like standard classification). That is, the response variable Z_i is a subset of the full label space $\mathcal{Y}$. For example, in the above in-silico screening task, X_i would be the current property being screened for, $\mathcal{Y}$ the space of all molecular candidates that might have this property, and $Z_i \subseteq \mathcal{Y}$ the set of molecules that do have it. Let $X_{n+1} \in \mathcal{X}$ be a new exchangeable test example for which we would like to predict the set of correct labels, $Z_{n+1} \subseteq \mathcal{Y}$. Our goal is to construct a set predictor $\mathcal{C}_k(X_{n+1})$ that maximizes recall of Z_{n+1} (i.e., TPR), while limiting the expected number of false

[1]CSAIL, Massachusetts Institute of Technology. [2]Google Research. Correspondence to: Adam Fisch <fisch@csail.mit.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

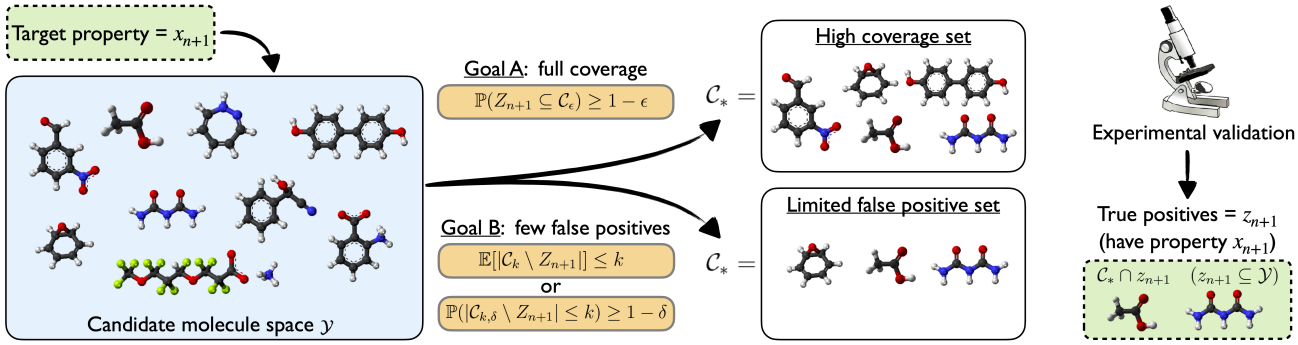


Figure 1: A demonstration of our approach to relaxing standard coverage guarantees ("Goal A") in favor of rigorous limits on the total number of false positives included in the output $\mathcal{C}_{k,\delta}$ ("Goal B"). In the illustrative case of in-silico screening for drug discovery, limiting false positives is critical when balancing a budget for experimental validation.

positives according to a user-defined tolerance $k \in \mathbb{R}_{>0}$:

$$\begin{aligned} &\text{maximize } \mathbb{E}\left[\frac{|\mathcal{C}_k(X_{n+1}) \cap Z_{n+1}|}{\max(|Z_{n+1}|, 1)}\right] \\ &\text{s.t. } \mathbb{E}\Big[|\mathcal{C}_k(X_{n+1}) \setminus Z_{n+1}|\Big] \leq k. \end{aligned} \tag{1}$$

As an alternative to bounding the *expected number* of false positives, we can also seek a predictor $\mathcal{C}_{k,\delta}$ that has more direct control of the *probability of exceeding* k false positives:

$$\begin{aligned} &\text{maximize } \mathbb{E}\left[\frac{|\mathcal{C}_{k,\delta}(X_{n+1}) \cap Z_{n+1}|}{\max(|Z_{n+1}|, 1)}\right] \\ &\text{s.t. } \mathbb{P}\Big(|\mathcal{C}_{k,\delta}(X_{n+1}) \setminus Z_{n+1}| \leq k\Big) \geq 1 - \delta, \end{aligned} \tag{2}$$

where $\delta \in (0, 1)$ is another user-defined tolerance level. Both constructions define different, but useful, operating conditions; the first is more straightforward (e.g., for the general practitioner), while the second offers a finer, two-parameter level of control. Note that both constraints are marginal over the choice of calibration and test data.

In order to achieve the desired levels of false positive control, we present an approach that is based on *set classification*, combined with conformal calibration techniques (Shafer and Vovk, 2008; Papadopoulos, 2008; Alvarsson et al., 2021). Specifically, we use a set nonconformity measure $\mathcal{F}\colon \mathcal{X} \times 2^{\mathcal{Y}} \to \mathbb{R}$ to score candidate output sets, $\mathcal{S} \in 2^{\mathcal{Y}}$, for a given input $x \in \mathcal{X}$. Intuitively, a *high* nonconformity score (e.g., loss) should reflect the confidence that the candidate set might contain a *high* number of false positives, and vice-versa. We learn this function from separate multi-label classification training data. As enumerating and scoring all possible candidate sets is combinatorially hard, we instead adopt the nested conformal prediction strategy of Gupta et al. (2019), where we greedily construct prediction sets using a best-first strategy that adds top-ranked individual labels to a growing, nested output set $\mathcal{S}$. We stop when its nonconformity score, $\mathcal{F}(x, \mathcal{S})$, exceeds a calibrated threshold—that we find based on our desired false positive constraints. This greedy approach both allows us to scale to larger label spaces $\mathcal{Y}$ (i.e., where there are many candidate labels that choose from when composing the prediction set), and to leverage powerful theory for calibrating expectations of monotonic losses for nested set predictors (Gupta et al., 2019; Bates et al., 2020; Angelopoulos et al., 2021b).

In summary, our main contributions are as follows:

- A theoretical adaptation of conformal prediction that provides rigorous false positive control instead of coverage;
- A simple and effective strategy for constructing valid output sets with empirically high true positive rates;
- A demonstration of the practical utility of our framework across a range of diverse classification tasks.

2 Related work

Uncertainty estimation. A large body of work in estimating model uncertainty focuses on calibrating model-based conditional probabilities, $p_\theta(\hat{y}_{n+1}|x_{n+1})$, such that the accuracy, $y_{n+1} = \hat{y}_{n+1}$, is equal to the estimated probability (Brier, 1950; Murphy and Epstein, 1967; Niculescu-Mizil and Caruana, 2005; Kuleshov et al., 2018; Kumar et al., 2019; Vaicenavicius et al., 2019). In theory, these estimates could be used to create prediction sets with few false positives, but they are not always accurate (Guo et al., 2017; Ashukha et al., 2020; Hirschfeld et al., 2020). In a similar vein, Bayesian formalisms quantify uncertainty via computing the posterior predictive distribution over model parameters (Neal, 1996; Graves, 2011; Hernández-Lobato and Adams, 2015; Gal and Ghahramani, 2016). However, the quality of these methods can vary depending on the suitability of the presumed prior and on approximation error.

Conformal prediction. As introduced in §1, conformal prediction (Vovk et al., 2005) provides a finite-sample, distribution-free method for obtaining prediction sets $\mathcal{C}$ with guarantees on the event $\mathbf{1}\{Y_{n+1} \in \mathcal{C}(X_{n+1})\}$. Most ef-

forts in CP focus on improving the predictive efficiency, $\mathbb{E}[|\mathcal{C}(X_{n+1})|]$, of the conformal sets (Vovk et al., 2016; Sadinle et al., 2019; Romano et al., 2020; Angelopoulos et al., 2021c; Fisch et al., 2021a;b; Hoff, 2021). As coverage is guaranteed by design, improving efficiency will naturally lead to more precise sets with fewer false positives—but not to a specifiable level. Cauchois et al. (2021) develop a conformal approach to multi-label classification that can guarantee that the prediction set only contains true labels (i.e., FP = 0), but does not offer fine-grained control. Most relevant to our work, Bates et al. (2020) develop a flexible framework for controlling the risk, $\mathbb{E}[\mathcal{L}(Y, \mathcal{T}(X))]$, of a set-valued predictor $\mathcal{T}$ with an arbitrary loss function $\mathcal{L}$—as long the loss respects a monotonic *nesting* property, $\mathcal{S} \subset \mathcal{S}' \Rightarrow \mathcal{L}(\mathcal{S}) \geq \mathcal{L}(\mathcal{S}')$, for any two prediction sets $\mathcal{S}$ and $\mathcal{S}'$. The calibration strategy we use here for marginal expectations is based on an extension in Angelopoulos et al. (2021b). Recently, Angelopoulos et al. (2021a) proposed methods to rigorously control non-monotonic losses, including the related false discovery rate (FDR), which normalizes the number of false positives over the size of the prediction set. However, as most of our target applications have relatively few true positives, FDR control can lead to many empty predictions (making controlling total false positives a more natural fit for this work, see Appendix E). Finally, though we focus on conformal approaches, our methods are tightly connected to the broader literature surrounding distribution-free calibration (Vovk et al., 2004; 2015; Vovk and Petej, 2014; Gupta et al., 2019; 2020; Barber, 2020).

Multiple testing. Controlling the number of false positives/discoveries over a collection of hypothesis tests is well-studied (Dunn, 1961; Benjamini and Hochberg, 1995; Lehmann and Romano, 2005; Romano and Wolf, 2007). In fact, the objectives expressed in Eqs. (1) and (2) are established concepts in statistics—i.e., PFER, the per-family error rate, and k-FWER, the familywise error rate (Spjøtvoll, 1972; Romano and Wolf, 2007). Recently, FDR control has also been studied for outlier detection in a conformal inference setting (Bates et al., 2021). Classic approaches operate over p-values for each hypothesis test that have specific dependency structures (e.g., independent or positively dependent), or otherwise use more conservative corrections. Though similar, our multi-label setting is slightly different from standard multiple testing in that there is both (1) an unknown dependency structure between candidate labels for the same query, but also (2) an extra layer of exchangeability over the $n+1$ queries. Our approach is able to ignore (1) by leveraging (2) within a conformal calibration framework.

Selective classification. In selective classification (El-Yaniv and Wiener, 2010), models can abstain from answering. In particular, Geifman and El-Yaniv (2017) propose a strategy for finding classifiers with specific selective $0/1$ risks (i.e., the expected accuracy over *answered* examples). In our setting, this is analogous to controlling false positives using $k \approx 0$. If uncertain, the model would have to "abstain" by outputting an empty set. Our framework generalizes this behavior to other types of constraints for any positive k.

3 Background

We begin with a brief review of conformal prediction (see Shafer and Vovk, 2008). Here, and in the rest of the paper, upper-case letters (X) denote random variables; lower-case letters (x) denote constants, and script letters ($\mathcal{X}$) denote sets, unless otherwise specified. Proofs are in Appendix A.

Given a new example x, for every candidate label $y \in \mathcal{Y}$ standard conformal classification (where there is one correct output) either accepts or rejects the null hypothesis that the pairing (x, y) is correct. The test statistic for this test is a *nonconformity measure*, $\mathcal{M}((x, y), \mathcal{D})$, where $\mathcal{D}$ is a dataset of exchangeable, labeled examples (as is (x, y_{true})). Informally, a lower value of $\mathcal{M}$ reflects that point (x, y) "conforms" to $\mathcal{D}$, whereas a higher value of $\mathcal{M}$ reflects that (x, y) is atypical relative to $\mathcal{D}$. A practical choice for $\mathcal{M}$ could be a model-based loss, e.g., $-\log p_\theta(y|x)$, where θ is a model fit to $\mathcal{D}$. For conformal prediction to work, it is important to ensure that $\mathcal{M}$ preserves the exchangeability over $\mathcal{D} \cup (x, y_{\text{true}})$. One such way is to learn $\mathcal{M}$ on separate data. Split conformal prediction (Papadopoulos, 2008) uses a proper training set $\mathcal{D}_{\text{train}}$ to learn a fixed $\mathcal{M}$ that is not modified during calibration or prediction. This trivially preserves exchangeability of the calibration and test points, and is a computationally efficient strategy (which we follow).

To construct a prediction set for the new test point x, the conformal classifier outputs all y for which the null hypothesis (that pairing (x, y) is correct) is not rejected. This is achieved by comparing the scores of the test candidate pairs to the scores computed over n calibration examples.

Theorem 3.1 (Split CP, Vovk et al. (2005); Papadopoulos (2008))**.** *Assume that examples $(X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, n+1$ are exchangeable. For a fixed nonconformity measure $\mathcal{M}$, let random variable $V_i = \mathcal{M}(X_i, Y_i)$ be the nonconformity score of (X_i, Y_i). For $\epsilon \in (0, 1)$, define the prediction (based on the first n examples) at $x \in \mathcal{X}$ as*

$$\mathcal{C}_\epsilon(x) := \tag{3}$$
$$\left\{y \in \mathcal{Y} \colon \mathcal{M}(x, y) \leq \text{Quantile}(1-\epsilon;\, V_{1:n} \cup \{\infty\})\right\}.$$

Then $\mathcal{C}_\epsilon(X_{n+1})$ satisfies $\mathbb{P}(Y_{n+1} \in \mathcal{C}_\epsilon(X_{n+1})) \geq 1 - \epsilon$.

Remark 3.2. Cauchois et al. (2021) extend the single label conformal prediction formulation to the multi-label case, where $Z_{n+1} \subseteq \mathcal{Y}$, by predicting two sets $\mathcal{C}_\epsilon^{\text{inner}}, \mathcal{C}_\epsilon^{\text{outer}} \subseteq \mathcal{Y}$ that fully sandwich Z_{n+1}, i.e., they guarantee $\mathbb{P}(\mathcal{C}_\epsilon^{\text{inner}}(X_{n+1}) \subseteq Z_{n+1} \subseteq \mathcal{C}_\epsilon^{\text{outer}}(X_{n+1})) \geq 1 - \epsilon$.

The motivation for our work is evident from Eq. (3): if we are unable to reject most candidates based on their nonconformity scores, then $\mathcal{C}_\epsilon$ can contain many false positives.

4 Set predictions with limited false positives

We now introduce our strategy for limiting the number of false positives that are contained in our output sets. To briefly remind the reader of our setting, we assume that we have been given n exchangeable multi-label classification examples, $(X_i, Z_i) \in \mathcal{X} \times 2^{\mathcal{Y}}$, $i = 1, \dots n$ as calibration data, that are drawn from a distribution P_{XZ}. We follow split conformal prediction, and assume that any training data used is distinct from this calibration data. The response Z_i is treated as a generalized set of correct labels for input X_i, and is a subset of $\mathcal{Y}$. For example, in the in-silico screening task from §1, X_i is the current target property being screened for, $\mathcal{Y}$ is the space of all molecular candidates, and $Z_i \subseteq \mathcal{Y}$ is the set of molecules that have that property.

For a prediction set $\mathcal{C}(x) \subseteq \mathcal{Y}$ evaluated at a point $x \in \mathcal{X}$ with label set $z \subseteq \mathcal{Y}$, we define the *true positive proportion* (TPP) as the ratio of correct labels that are recovered:

$$\mathrm{TPP}(z, \mathcal{C}(x)) := \frac{|\mathcal{C}(x) \cap z|}{\max(|z|, 1)} \tag{4}$$

(note that $\mathrm{TPR} := \mathbb{E}[\mathrm{TPP}]$), and the number of *false positives* (FP) as the total count of incorrect labels in $\mathcal{C}(x)$:

$$\mathrm{FP}(z, \mathcal{C}(x)) := |\mathcal{C}(x) \setminus z|. \tag{5}$$

Our goal, as stated in §1, is to maximize the expected TPP, while constraining the FP in either of two ways:

Definition 4.1 (k-FP validity)**.** A conformal classifier producing random test prediction $\mathcal{C}_k(X_{n+1})$ is k-FP valid if it satisfies $\mathbb{E}[\mathrm{FP}(Z_{n+1}, \mathcal{C}_k(X_{n+1}))] \leq k$.

Definition 4.2 ((k, δ)-FP validity)**.** A conformal classifier producing random test prediction $\mathcal{C}_{k,\delta}(X_{n+1})$ is (k, δ)-FP valid if it satisfies $\mathbb{P}(\mathrm{FP}(Z_{n+1}, \mathcal{C}_{k,\delta}(X_{n+1})) \leq k) \geq 1 - \delta$.

4.1 An oracle set predictor

To motivate our approach, imagine an *oracle* with access to $P_{Z|X}$, the conditional distribution of the multi-label set Z given the input X. Given this information, for any input $x \in \mathcal{X}$ and candidate set $\mathcal{S} \in 2^{\mathcal{Y}}$, in theory such an oracle would be able to exactly calculate both the expectation and the conditional distribution of the number of false (and true) positives in $\mathcal{S}$ given x. In order to maximize the TPR while meeting k-FP and (k, δ)-FP validity, it could then yield:

$$\mathcal{C}_k^{\mathrm{oracle}}(x) := \tag{6}$$
$$\operatorname*{arg\,max}_{\mathcal{S} \in 2^{\mathcal{Y}}} \left\{ \mathbb{E}[\mathrm{TPP}(Z, \mathcal{S}) \mid x] \colon \mathbb{E}[\mathrm{FP}(Z, \mathcal{S}) \mid x] \leq k \right\}$$

$$\mathcal{C}_{k,\delta}^{\mathrm{oracle}}(x) := \tag{7}$$
$$\operatorname*{arg\,max}_{\mathcal{S} \in 2^{\mathcal{Y}}} \left\{ \mathbb{E}[\mathrm{TPP}(Z, \mathcal{S}) \mid x] \colon \mathbb{P}(\mathrm{FP}(Z, \mathcal{S}) \mid x] > k) < \delta \right\}$$

where ties are settled by smaller set size. Of course, computing this oracle is not possible, as $P_{Z|X}$ is unknown. Furthermore, enumerating all sets $\mathcal{S} \in 2^{\mathcal{Y}}$ is infeasible for large $\mathcal{Y}$. Instead, in the following sections we develop a practical approach for roughly approximating the oracle's behavior with three main components:

1. A **set function** $\mathcal{F} \colon \mathcal{X} \times 2^{\mathcal{Y}} \to \mathbb{R}$ that directly generates a score for a candidate set $\mathcal{S}$ given x that is predictive of either $\mathbb{E}[\mathrm{FP}(Z, \mathcal{S}) \mid x]$ or $\mathbb{P}(\mathrm{FP}(Z, \mathcal{S}) \mid x] > k)$.
2. A **calibrated search strategy** for exploring a tractable number of candidate sets, and identifying valid sets satisfying our constraints using predictions from $\mathcal{F}$;
3. A **selection policy** for picking a final output set.

Wherever possible, our proposed method will try to balance simplicity and efficiency with effectiveness. Theoretically, however, the framework it follows is model-agnostic.

4.2 Scoring candidate sets with set functions

We choose to model $\mathcal{F}$ using DeepSets (Zaheer et al., 2017). DeepSets is a popular method which is known to be a universal approximator for continuous set functions, which makes it a natural choice for our purpose. Let $\{\phi(x, y_1), \dots, \phi(x, y_{|\mathcal{S}|})\}$ featurize a candidate set $\mathcal{S} \subseteq \mathcal{Y}$, where $\phi(x, y_c) \in \mathbb{R}^d$ is a function of (x, y_c), for $y_c \in \mathcal{S}$. In practice, we find that taking $\phi(x, y_c)$ to be an estimate of $p_\theta(y_c \in Z \mid x)$, the marginal likelihood of y_c being a correct label, performs well and is simple to implement. These one-dimensional prediction scores can be provided by any base model.[1] For example, in our in-silico screening task, we define ϕ using a directed MPNN (Yang et al., 2019) that independently predicts the probability of an individual molecule having the properties targeted by the screen, or not. Given ϕ, the DeepSets model is defined by

$$\Psi(x, \mathcal{S}) := \mathrm{softmax}\Big(\mathrm{dec}\Big(\sum_{y_c \in \mathcal{S}} \mathrm{enc}(\phi(x, y_c))\Big)\Big), \tag{8}$$

where $\mathrm{enc}(\cdot)$ and $\mathrm{dec}(\cdot)$ are neural encoder and decoder models, and $\mathrm{softmax}(\cdot)$ is taken over the range of possible false positives, $\{0, \dots, |\mathcal{S}|\}$. Ψ is trained to predict the total number of false positives in $\mathcal{S}$ via cross entropy, using labeled sets sampled from held-out training data, separate from the split used to learn p_θ (used for ϕ). We then compute $\mathcal{F}_k$ and $\mathcal{F}_{k,\delta}$ (for either k-FP or (k, δ)-FP validity) as

$$\mathcal{F}_k(x, \mathcal{S}) := \sum_{\eta=0}^{|\mathcal{S}|} \eta \cdot \Psi(x, \mathcal{S})_\eta \tag{9}$$

$$\mathcal{F}_{k,\delta}(x, \mathcal{S}) := 1 - \sum_{\eta=0}^{\min(k, |\mathcal{S}|)} \Psi(x, \mathcal{S})_\eta, \tag{10}$$

where $\Psi(x, \mathcal{S})_\eta$ denotes the η-th index of the $\mathrm{softmax}$ (i.e., the estimated probability that FP = η). Additional details on

[1] This is comparable to the 1-d features used by Platt scaling.

Algorithm 1 Pseudocode for conformal prediction with limited false positives (in expectation case, see Eq. (1)).

Definitions: x_{n+1} is a test point, $\mathcal{D}_{\text{train}}$ is a training set, $\mathcal{D}_{\text{cal}}$ is a calibration set, k is the tolerance, and B is a parameter for considering only the top individually ranked candidates, $y_c \in \mathcal{Y}$. LikelihoodModel is an abstract model that estimates individual label likelihood for ranking and set item featurization. SetModel is an abstract model that estimates FP (we use DeepSets).

1: # Using a training set, fit both a LikelihoodModel p_θ and a SetModel $\mathcal{F}$ (§4.2).
2: **function** TRAIN($\mathcal{D}_{\text{train}}$)
3: $\quad\mathcal{D}^{(1)}_{\text{train}}, \mathcal{D}^{(2)}_{\text{train}} \leftarrow$ SPLIT($\mathcal{D}_{\text{train}}$) # Split the training data into two disjoint sets.
4: $\quad p_\theta(y_c \in Z \mid x) \leftarrow$ FIT(LikelihoodModel, $\mathcal{D}^{(1)}_{\text{train}}$) # Use one set to estimate individual likelihoods, $p_\theta(y_c \in Z \mid x)$.
5: $\quad\mathcal{F}(x, \mathcal{S}) \leftarrow$ FIT(SetModel, $p_\theta, \mathcal{D}^{(2)}_{\text{train}}$) # Use the other (smaller) set to learn the FP set function, $\mathcal{F}(x, \mathcal{S})$.
6: $\quad$**return** $p_\theta, \mathcal{F}$
7: **end function**
8: # Using the trained p_θ and $\mathcal{F}$ models, find a set score threshold t_k on a calibration set that achieves k-FP validity (§4.3).
9: **function** CALIBRATE(p_θ, $\mathcal{F}$, $\mathcal{D}_{\text{cal}}$, k, B)
10: $\quad\mathcal{T}_{\text{cal}} = \{\}$
11: $\quad$**for** $(x_i, z_i) \in \mathcal{D}_{\text{cal}}$ **do**
12: $\quad\quad\{y_{i,\pi(1)}, \ldots, y_{i,\pi(B)}\} \leftarrow \text{SORT}(\mathcal{Y}, p_\theta(y_c \in Z_i \mid x_i))_{1:B}$ # Rank top B candidates by individual likelihood.
13: $\quad\quad\{\mathcal{S}_{i,1}, \ldots, \mathcal{S}_{i,B}\} \leftarrow \{y_{i,\pi(1:j)} : j \in \{1, \ldots, B\}\}$ # Construct nested sets using this ordering.
14: $\quad\quad\{v_{i,1}, \ldots, v_{i,B}\} \leftarrow \{\mathcal{F}(x_i, \mathcal{S}_{i,1}), \ldots \mathcal{F}(x_i, \mathcal{S}_{i,B})\}$ # Compute nonconformity scores using $\mathcal{F}$.
15: $\quad\quad\text{FP}_{\max}(x_i, z_i, t) \leftarrow \text{CACHE}(x_i, z_i, v_{i,1:B}, \mathcal{S}_{i,1:B})$ # Cache dependent variables for $\text{FP}_{\max}(x_i, z_i, t)$.
16: $\quad\quad\mathcal{T}_{\text{cal}} \leftarrow \mathcal{T}_{\text{cal}} \cup \{\text{FP}_{\max}(x_i, z_i, t)\}$ # Append cached $\text{FP}_{\max}(x_i, z_i, t)$ to the calibration set.
17: $\quad$**end for**
18: $\quad t_k \leftarrow \text{FIND\_THRESHOLD}(\mathcal{T}_{\text{cal}}, B, k)$ # Use Eq. (15) to find a k-FP valid set score threshold.
19: $\quad$**return** t_k
20: **end function**
21: # Using trained p_θ and $\mathcal{F}$ models and calibrated threshold t_k, return a TPR-maximizing prediction set for test point x_{n+1} (§4.4).
22: **function** PREDICT(x_{n+1}, p_θ, $\mathcal{F}$, t_k, B)
23: $\quad$# Repeat lines 12-14 to compute $\mathcal{S}_{n+1,1:B}$ and $v_{n+1,1:B}$.
24: $\quad\mathcal{J} \leftarrow \{j \in \{1, \ldots, B\} : v_{n+1,j} < t_k\}$ # Identify indices of candidate sets that pass threshold t_k.
25: $\quad\mathcal{C}_k(x_{n+1}) \leftarrow \mathcal{S}_{n+1,\max \mathcal{J}}$ # Choose the largest sized set among filtered candidates.
26: $\quad$**return** $\mathcal{C}_k(x_{n+1})$
27: **end function**

how to train $\mathcal{F}$ are given in Appendix B. In the next sections, we will now only refer to $\mathcal{F}$ as a general function.

4.3 Searching for valid candidate sets

Although our set predictor $\mathcal{F}$ is trained to model either the expected FP or its CDF, it is not necessarily accurate. If $\mathcal{F}$ were simply substituted into Eq. (6) or Eq. (7), it may not produce valid set predictions. To account for this mismatch, we must carefully calibrate a threshold for accepting candidate sets based on $\mathcal{F}$. At the same time, we also must efficiently search the combinatorial space of candidate sets.

To efficiently calibrate our predictor, we cast our approach into a form of *nested* conformal prediction (Gupta et al., 2019). First, we greedily identify a sequence of nested candidate sets, $\varnothing \subset \mathcal{S}_1 \subset \mathcal{S}_2 \subset \ldots \subset \mathcal{S}_j$, by ranking individual labels $y_c \in \mathcal{Y}$ according to some auxiliary model, and including them one by one into the growing output set $\mathcal{S}_{j+1}$. Notice that, by construction, the number of false positives contained in $\mathcal{S}_j$ is non-decreasing in index j, i.e.,

$$j \leq j' \implies \text{FP}(z, \mathcal{S}_j) \leq \text{FP}(z, \mathcal{S}_{j'}). \tag{11}$$

In practice, we find that ranking individual labels by their estimated marginal likelihoods of being true positives, i.e., $p_\theta(y_c \in Z \mid x)$—the same model used in §4.2—performs well and avoids the overhead of training an additional scoring model. Importantly, for further efficiency (elaborated on in Remark 4.5) we only consider sets up to a maximum size $B \leq |\mathcal{Y}|$, where B is a hyper-parameter that we can set.

Next, we compute a set nonconformity score v_j (assumed to be finite) for each candidate set $\mathcal{S}_j$ using $\mathcal{F}$, where

$$v_j := \mathcal{F}(x, \mathcal{S}_j). \tag{12}$$

Finally, we define the worst-case number of false positives over all nested candidate sets $\mathcal{S}_{1:B}$ having nonconformity scores less than t (given the input x with label set z) as

$$\text{FP}_{\max}(x, z, t) := \max \big\{\text{FP}(z, \mathcal{S}_j) : v_j < t\big\}. \tag{13}$$

If this set is empty, then $\text{FP}_{\max}$ is 0. Due to our nested construction, this is also simply the number of false positives contained in the largest set $\mathcal{S}_j$ satisfying $v_j < t$. It is simple to show that $\text{FP}_{\max}$ is non-decreasing in t, as stated below.

Lemma 4.3 (Monotonicity). *For sets $\mathcal{S}_j$ and scores v_j and $\text{FP}_{\max}(x, z, t)$ as defined in Eqs. (12) and (13), respectively,*

$$t \leq t' \implies \text{FP}_{\max}(x, z, t) \leq \text{FP}_{\max}(x, z, t'). \tag{14}$$

Dataset	Input	# Examples	# Negatives	# Positives	% Empty
In-silico screening	SMILES	5,000	85 (50-97)	15 (3-50)	0.0
Object detection	Image	3,000	96 (89-98)	4 (2-11)	1.1
Entity extraction	Text	3,453	99 (97-100)	1 (0-3)	20.2

Table 1: Dataset statistics (test split). Numbers are reported with respect to the top $B = 100$ candidates per example. The median number of positives and negatives per example is given, in addition to their 16th and 84th percentiles. We also give statistics for the percentage of examples that have "empty" label sets with no positives (i.e., the label set has $|z| = 0$).

Using this key property, we can find a *maximal* threshold t to use as a "cutoff point" for the sequence of nested candidate sets such that $\mathrm{FP}_{\max}$ is controlled, as formalized next.

Theorem 4.4 (FP-CP)**.** *Assume that examples $(X_i, Z_i) \in \mathcal{X} \times 2^{\mathcal{Y}}$, $i = 1, \ldots, n+1$ are exchangeable. For each example i, let $S_{i,j}$, $j = 1, \ldots, B$ (where $B \leq |\mathcal{Y}|$ is a finite hyper-parameter) be candidate sets, where finite random variable $V_{i,j} = \mathcal{F}(X_i, \mathcal{S}_{i,j})$ is a set nonconformity score. For tolerances $k \in \mathbb{R}_{>0}$ and $\delta \in (0, 1)$ define the random variables T_k and $T_{k,\delta}$ (based on the first n examples) as*

$$T_k := \sup\left\{t \in \mathbb{R}\colon \frac{B + \sum_{i=1}^{n} \mathrm{FP}_{\max}(X_i, Z_i, t)}{n+1} \leq k\right\} \quad \text{and} \tag{15}$$

$$T_{k,\delta} := \sup\left\{t \in \mathbb{R}\colon \frac{\sum_{i=1}^{n} \mathbf{1}\{\mathrm{FP}_{\max}(X_i, Z_i, t) \leq k\}}{n+1} \geq 1 - \delta\right\}, \tag{16}$$

where $\mathrm{FP}_{\max}$ is as defined in Eq. (13). Then we have that

$$\mathbb{E}\Big[\mathrm{FP}_{\max}(X_{n+1}, Z_{n+1}, T_k)\Big] \leq k, \quad \text{and} \tag{17}$$

$$\mathbb{P}\Big(\mathrm{FP}_{\max}(X_{n+1}, Z_{n+1}, T_{k,\delta}) \leq k\Big) \geq 1 - \delta. \tag{18}$$

Remark 4.5. The hyper-parameter B plays an important role when controlling for k-FP. T_k may be very conservative if $B = |\mathcal{Y}|$ and $|\mathcal{Y}|$ is very large, to the point where $T_k = -\infty$ always if $|\mathcal{Y}| > k(n+1)$. It can therefore be beneficial to truncate the considered label space $\mathcal{Y}$ for an example x to only the top $B \ll k(n+1)$ individual candidates, $\{y_1, \ldots, y_B\} \in \mathcal{Y}^B$. For example, for text generation tasks (like machine translation), $\mathcal{Y}$ is infinite, but we can restrict our predictions to a subset of the top B beam search candidates (where B can still be reasonably large). Still, this isn't free: a smaller B may result in fewer true positives.

Remark 4.6. No constraints are placed on the underlying set function $\mathcal{F}$ in Theorem 4.4; i.e., it need not be a DeepSets architecture. If, however, $\mathcal{F}$ is a good estimator of $\mathrm{FP}(Z, \mathcal{S}) \mid X$, then our method is more likely to identify sets that are approximately valid conditioned on $X_{n+1} = x_{n+1}$, which we investigate empirically in §6.

Remark 4.7. It is useful to note that nestedness of $\mathcal{S}_{i,j}$ is not necessary for the above calibration to hold (it is mainly used for efficiency). Monotonicity of $\mathrm{FP}_{\max}$ is sufficient.

4.4 Selecting the final output set

The main consequence of Theorem 4.4 is that, using the calibrated nonconformity threshold T_k or $T_{k,\delta} = t^*$, we can construct a collection of sets that are simultaneously valid by keeping all candidate sets with scores less than t^*. Specifically, we are free to select *any* set in the filtered set of candidates $S_{n+1,j}$, $j \in \mathcal{J}$ where $\mathcal{J} := \{j\colon v_{n+1,j} < t^*\}$, as a valid output. Ideally, we would be able to follow the the oracle strategy in returning the smallest set with the highest number of true positives. This would make our predictions *efficient*, in the sense that we are not including more false positives than necessary (even if the total is still $\leq k$). A reasonable choice is to then choose $\mathcal{S}_{j^*}$ where $j^* := \arg\max_{j \in \mathcal{J}} |\mathcal{S}_j| - \mathcal{F}(x, \mathcal{S}_j)$; but this can be sub-optimal if $\mathcal{F}$ is not accurate. As a greedy, but effective, approach we simply take the *largest* set as our final output, which has maximal TPR. We formalize this in Proposition 4.8.

Proposition 4.8 (Greedy FP-CP)**.** *Let $T_\circ$ denote either T_k or $T_{k,\delta}$. Then random candidate sets $\mathcal{S}_{n+1,j}$, $\forall j \in \mathcal{J} := \{j\colon V_{n+1,j} < T_\circ\}$, are valid. Furthermore, among indices $\mathcal{J}$, $\max \mathcal{J}$ indexes the nested set with the highest TPR.*

We discuss some additional considerations of our method, as well as potential limitations and extensions, in Appendix E.

5 Experimental setup

In this section, we outline our tasks and models. We also describe our evaluation and baselines. For all experiments, we set B to 100. Table 1 provides statistics for the datasets used in experiments. Appendix C contains additional details.

5.1 Tasks

In-silico screening for drug discovery. As introduced in §1, the goal of in-silico screening is to identify potentially effective drugs to manufacture and test. We use the ChEMBL database (Mayr et al., 2018) to screen molecules for combinatorial constraint satisfaction, where given a constraint such as "*has property A but not property B*," we want to identify the subset of molecules from a given set of candidates that have the desired attributes. We partition the dataset both by molecules and property combinations, so that at test time the model makes predictions on combinations it has never been tested on before (after being trained on the same

properties, but seen only in different combinations), over a pool of molecules that it has never seen before. Scores for candidate molecules are obtained via an ensemble of directed MPNNs (Yang et al., 2019).

Object detection. We consider the task of placing bounding boxes around all objects of a certain type (such as a person) that are present in an image (of which there may be few, many, or none). We use the MS-COCO dataset (Lin et al., 2014), a dataset with images of everyday scenes containing 80 object types (e.g., person, bicycle, dog, car, etc). We extract typed bounding box candidates (i.e., tuples of both location *and* category) using an EfficientDet model (Tan et al., 2020) with non-maximum suppression. True positives are defined as boxes that have an intersection over union (IoU) > 0.5 with a matching annotation of the same type.

Entity extraction. In entity extraction, we are interested in identifying all named entities that appear in a tokenized sentence x of length l, where $x = \{w_1, \ldots, w_l\}$, and classifying them into appropriate categories. A named entity is a proper noun, demarcated by a contiguous span $\{w_{\text{start}}, \ldots, w_{\text{end}}\} \subseteq x$ of the input sentence, that can be associated with a particular class of interest (such as a person, location, organization, or product). We report results on the CoNLL NER dataset (Tjong Kim Sang and De Meulder, 2003), where we use the PURE span-based entity extraction model of Zhong and Chen (2021) to individually score all $\mathcal{O}(l^2)$ candidate spans. We consider exact span predictions of the correct category to be true positives, and all others to be false positives. Many sentences contain no entities.

5.2 Evaluation

For each task we learn all models on a training set, perform model selection on a validation set, and report final results as the average over 1000 random trials on a test set, where in each trial we partition the data into 80% calibration ($x_{1:n}$) and 20% prediction points (x_{n+1}). To compare across k, we plot each metric as a function of k (up to $k = B$), and compute the area under the curve (AUC). Shaded regions show the 16-84th percentiles across trials. In addition to TPR (our main metric), as our method already guarantees marginal FP-validity, we also compute the size-stratified k-FP (SSFP_k) and (k, δ)-FP ($SSFP_{k,\delta}$) violation (Angelopoulos et al., 2021c), see Appendix C.1. Lower size-stratified violation suggests that a classifier has better conditional coverage. We also report average FP results in Table 2.

5.3 Baselines

For all experiments, we compare our FP-CP (NN) method using a DeepSets-based $\mathcal{F}$ to the following baselines:

1. **Top-k.** We naively take the top k' fixed predictions for any x_{n+1}, where k' is found using average performance on the calibration set (without any correction factors, so it is *not* guaranteed to be valid). Note that k' can be (and mostly is) different than the user-specified k for FP.
2. **Outer Sets @ 90.** We use the (one-sided) multi-label conformal prediction technique of Cauchois et al. (2021) to bound $\mathbb{P}(Z_{n+1} \subseteq \mathcal{C}_\epsilon(X_{n+1})) \geq 0.90$. Though not directly comparable, we use this to benchmark our method against sets that preserve marginal coverage (at a typical level). For simplicity, we use the direct inner/outer method without dynamic CQC quantiles.[2]
3. **Inner Sets.** Again, we use the (one-sided) method of Cauchois et al. (2021), this time to bound $\mathbb{P}(\mathcal{C}_\epsilon(X_{n+1}) \subseteq Z_{n+1}) \geq 1 - \epsilon$ at level $\epsilon = k/B$ (recall that $B \leq |\mathcal{Y}|$ is the truncation parameter, and the FP upper bound) for k-FP control and at level $\epsilon = \delta$ for (k, δ)-FP control. It is straightforward to show that these levels of ϵ conservatively achieve k-FP and (k, δ)-FP control.
4. **Independent scoring (max).** We take $\mathcal{F}(x, \mathcal{S})$ to be the maximum individual label uncertainty in $\mathcal{S}$, $\max\{1 - p_\theta(y_c \in Z \mid x) \colon y_c \in \mathcal{S}\}$. This is equivalent to choosing labels independently. Calibration uses the same FP-CP algorithm (it is a drop-in replacement for the NN).
5. **Cumulative scoring (sum).** We take $\mathcal{F}(x, \mathcal{S})$ to be the cumulative individual label uncertainty in $\mathcal{S}$, $\sum_{y_c \in \mathcal{S}} 1 - p_\theta(y_c \in Z \mid x)$. We calibrate $p_\theta(y_c \in Z \mid x)$ using Platt scaling (Platt, 1999). As with the max scoring baseline, calibration uses the same FP-CP algorithm.

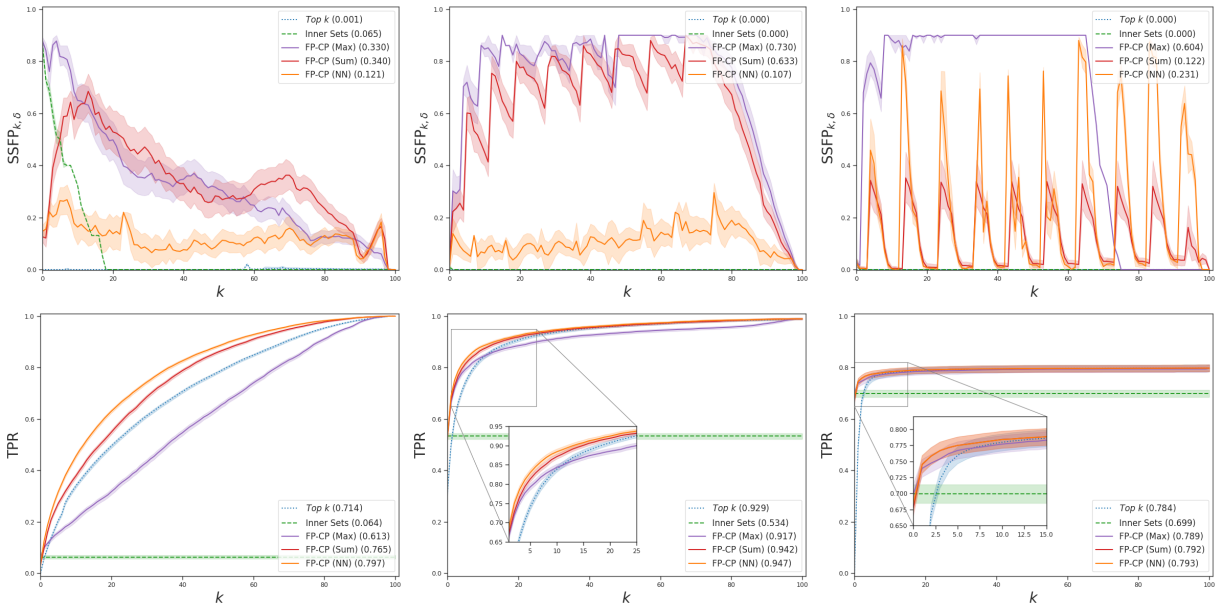
Baseline (1) contrasts our approach with what is normally a "first thought" in practice, (2) and (3) test the efficacy of our system over existing techniques, and (4) and (5) demonstrate our FP-CP calibration with simpler alternatives for $\mathcal{F}$.

6 Experimental Results

We now present our empirical results. Figure 3 and Figure 2 present AUC results, computed over all values of ϵ, for all tasks. Table 2 reports additional absolute results for a number of reference k values, focusing on the in-silico screening task. Appendix D contains additional discussion.

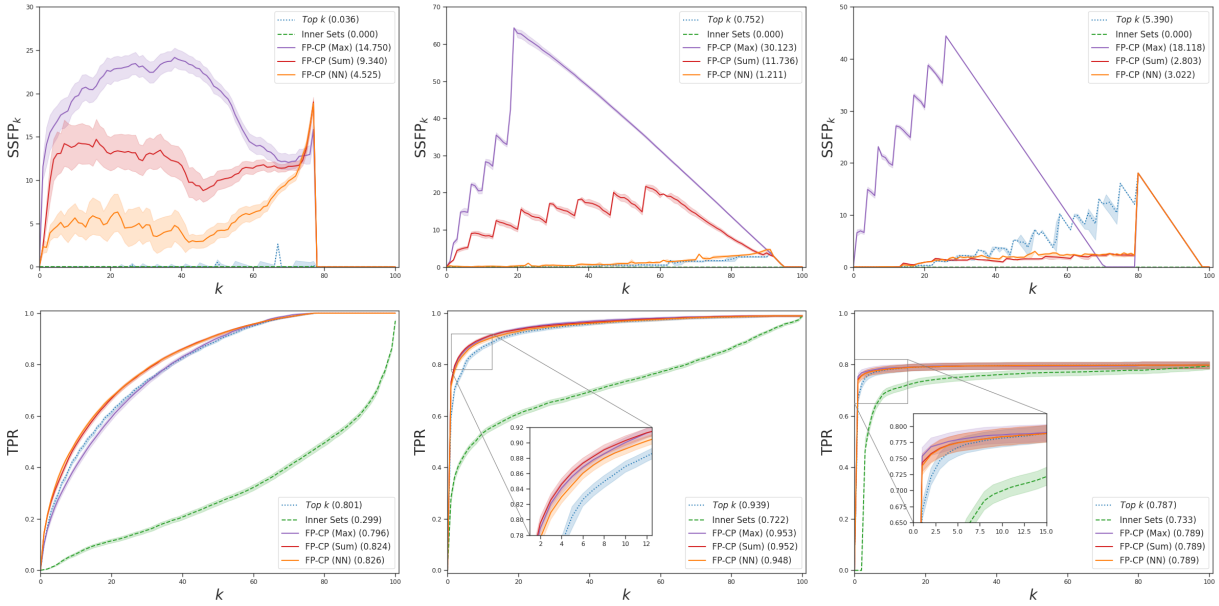
Limiting false positives. The top rows of Figures 2 and 3 show the size-stratified violation for (k, δ)-FP and k-FP, respectively. Across values of k, FP-CP (NN) typically achieves substantially *lower* worst-case violations than either max or sum scoring alternatives, (though, in some cases, the magnitude of SSFP can depend strongly on k). The Top-k and Inner Sets approaches also prevent large violations (though, by itself, this result is not necessarily impressive, as always returning an empty set will lead to $\text{SSFP} = 0$). When accounting for TPR (bottom rows), we see that our FP-CP methods demonstrate stronger performance.

[2] Preliminary experiments indicated that including CQC quantiles did not lead to significantly different (marginal) results.



(a) In-silico screening (b) Object detection (c) Entity extraction

Figure 2: (k, δ)-FP results as a function of k for $\delta = 0.1$ up to $k = B = 100$. The top row plots $\mathrm{SSFP}_{k,\delta}$ violation (lower is better). The bottom row plots TPR (higher is better). We see that compared to the other baselines, our conformal DeepSets approach (NN) has the best (or close to) TPR AUC across tasks, while having the lowest (or close to) $\mathrm{SSFP}_{k,\delta}$ violation.



(a) In-silico screening (b) Object detection (c) Entity extraction

Figure 3: k-FP results as a function of k up to $k = B = 100$. The top row plots size-stratified k-FP violation (lower is better). The bottom row plots the TPR (higher is better). As k grows, our methods quickly achieve high TPR. Consistent with Figure 2, our conformal DeepSets approach (NN) demonstrates high TPR and low SSFP_k across tasks.

Maximizing true positive rates. The bottom rows of Figures 2 and 3 plot TPR rates and AUC across values of k, while Table 2 details results for several representative individual configurations. On the screening task, we see that our FP-CP (NN) method provides significantly higher TPR than other baselines. For example, allowing no more than 5 false positives leads to a TPR of 36.1% with k-FP. In comparison, the TPR of Top-k is only 29.8%. As might be expected, the advantage of the DeepSets approach underlying FP-CP (NN) over simpler FP-CP scoring mechansims

	Top k		Inner Sets		FP-CP (Max)		FP-CP (Sum)		FP-CP (NN)	
k-FP:										
	Avg. FP	*TPR*	*Avg.* FP	*TPR*	*Avg.* FP	*TPR*	*Avg.* FP	*TPR*	*Avg.* FP	*TPR*
$k = 5$	4.59	29.8	0.14	2.5	4.98	27.5	4.99	34.1	4.98	36.1
$k = 15$	14.47	53.4	0.88	9.5	14.98	50.7	14.99	58.8	14.99	59.9
$k = 25$	24.51	68.0	1.49	13.4	24.98	66.8	24.99	73.1	24.99	73.2
$k = 35$	34.54	78.2	2.45	18.4	34.97	78.4	34.99	82.6	34.99	82.5
(k, δ)-*FP with* $1 - \delta = 0.9$:										
	FP $\leq k$	*TPR*	FP $\leq k$	*TPR*	FP $\leq k$	*TPR*	FP $\leq k$	*TPR*	FP $\leq k$	*TPR*
$k = 5$	100.0	20.5	96.6	6.36	90.0	15.8	90.0	27.2	90.0	31.6
$k = 15$	94.7	42.4	99.5	6.36	90.0	26.7	90.0	47.4	90.0	55.3
$k = 25$	96.6	55.7	100.0	6.36	90.0	37.4	90.0	62.3	90.0	69.0
$k = 35$	97.5	66.2	100.0	6.36	90.0	49.1	90.0	74.0	90.0	79.0

Table 2: Results for the in-silico screening task on the ChEMBL dataset. For k-FP validity, we report the empirical average of false positives in the prediction sets. For (k, δ)-FP validity we report the percentage of prediction sets with $\leq k$ false positives. TPR is expressed as a percent. Our FP-CP methods meet our target thresholds; using the Inner Sets approach does too, but is conservative (as expected). Applying FP-CP calibration with our DeepSets model (NN) yields substantially higher TPR across various tolerance levels compared to the other baseline scoring mechanisms.

Task	TPR	Avg. FP	Avg. Size
In-silico screening	97.2	63.6	86.6
Object detection	96.1	32.4	38.2
Entity extraction	75.0	0.77	2.31

Table 3: Outer Sets applied at coverage 0.90 for comparison. Note that as some examples do *not* have any positives, full coverage in the typical sense is not always achievable. Average FP and set size are reported as absolute values.

is more pronounced for tasks with higher cardinality label sets, such as in-silico screening versus object detection of entity extraction (see a comparison of dataset characteristics in Table 1). Furthermore, since entity extraction contains a high proportion of examples with "empty" label sets, we can see that its TPR asymptotes at the natural rate of answerable examples. Nevertheless, in general, all FP-CP methods (with max, sum, or NN scoring) provide high TPR (exceeding non FP-CP methods) even at low values of k.

Comparison to conformal coverage methods. Table 3 gives the results of the coverage-seeking Outer Sets method at level 0.90 (a typical tolerance). Indeed, we achieve strong TPR (97.2% for the in-silico screening task), but also incur a high false positive cost in the process (63.6 average FP for in-silico screening). In contrast, our method allows us to directly limit false positives, without losing high TPR empirically (e.g., equivalently controlling for ≤ 63.6 FP, we acheive 97.0% TPR on the in-silico screening task).

7 Conclusion

Conformal prediction, in its standard formulation, already grants theoretical performance guarantees that can be critical in many applications. Naively applying CP, however, can yield disappointing results. Even if the target coverage is upheld, the predicted sets may be too large, and too noisy, to be practical. In this paper, we proposed a method for trading coverage guarantees in favor of strict limits on the number of false positives contained in our prediction sets. Our results show that our method yields classifiers that (1) still achieve strong true positive rates compared to their coverage-seeking counterparts, and (2) predict meaningful output sets with effectively controlled numbers of false positives.

Acknowledgements

We thank Ben Fisch, Anastasios Angelopoulous, Stephen Bates, and Lihua Lei for valuable technical feedback and discussions. AF is supported in part by the NSF Graduate Research Fellowship. This work is also supported in part by the MLPDS Consortium and the DARPA AMD project.

Ethics Statement

Our FP control methods are general and can be applied to many applications and on top of any model for computing nonconformity scores. It is important to acknowledge that any undesirable biases exhibited by underlying models can still propagate to the prediction sets of our methods. While our methods provide marginal performance guarantees, we recommend that any application to perform controlled evaluation across target populations to ensure fairness.

Reproducibility Statement

All datasets and base models used in this paper are publicly available (see §5.1 and Appendix C for details). The results in Section 6 are based on the experimental setting described in Section 5. Our code is publicly available at https://github.com/ajfisch/conformal-fp.

References

Jonathan Alvarsson, Staffan Arvidsson McShane, Ulf Norinder, and Ola Spjuth. Predicting with confidence: Using conformal prediction in drug discovery. *Journal of Pharmaceutical Sciences*, 110(1):42–49, 2021.

Anastasios Nikolas Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *ArXiv preprint: 2110.01052*, 2021a.

Anastasios Nikolas Angelopoulos, Stephen Bates, and Lihua Lei. Risk control in expectation: A proof and counter example, 2021b. URL http://angelopoulos.ai/publications/downloads/rce_note.pdf.

Anastasios Nikolas Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I. Jordan. Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations (ICLR)*, 2021c.

Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Rina Foygel Barber. Is distribution-free inference possible for binary regression? *Electronic Journal of Statistics*, 14 (2):3487 – 3524, 2020.

Stephen Bates, Anastasios Nikolas Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution free, risk controlling prediction sets. *ArXiv preprint: 2101.02703*, 2020.

Stephen Bates, Emmanuel Candès, Lihua Lei, Yaniv Romano, and Matteo Sesia. Testing for outliers with conformal p-values. *ArXiv preprint: 2104.08279*, 2021.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 1995.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78:1–3, 1950.

Maxime Cauchois, Suyash Gupta, and John C. Duchi. Knowing what you know: valid and validated confidence sets in multiclass and multilabel prediction. *Journal of Machine Learning Research (JMLR)*, 2021.

Olive Jean Dunn. Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293), 1961.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research (JMLR)*, 11, 2010.

Adam Fisch, Tal Schuster, Tommi Jaakkola, and Regina Barzilay. Efficient conformal prediction via cascaded inference with expanded admission. In *International Conference on Learning Representations (ICLR)*, 2021a.

Adam Fisch, Tal Schuster, Tommi Jaakkola, and Regina Barzilay. Few-shot conformal prediction with auxiliary tasks. In *International Conference on Machine Learning (ICML)*, 2021b.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, 2016.

Alexander Gammerman and Vladimir Vovk. Hedging Predictions in Machine Learning: The Second Computer Journal Lecture . *The Computer Journal*, 50(2), 2007.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

David L. Gold, Jeffrey C. Miecznikowski, and Song Liu. Error control variability in pathway-based microarray analysis. *Bioinformatics*, 25(17):2216–2221, 2009.

Alex Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2011.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, 2017.

Chirag Gupta, Arun K. Kuchibhotla, and Aaditya K. Ramdas. Nested conformal prediction and quantile out-of-bag ensemble methods. *arXiv: 1910.10562*, 2019.

Chirag Gupta, Aleksandr Podkopaev, and Aaditya Ramdas. Distribution-free binary classification: prediction sets, confidence intervals and calibration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

José Miguel Hernández-Lobato and Ryan P. Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning (ICML)*, 2015.

Lior Hirschfeld, Kyle Swanson, Kevin Yang, Regina Barzilay, and Connor W. Coley. Uncertainty quantification using neural networks for molecular property prediction. *ArXiv preprint: 2005.10036*, 2020.

Peter Hoff. Bayes-optimal prediction with frequentist coverage control. *ArXiv preprint: 2105.14045*, 2021.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning (ICML)*, 2018.

Ananya Kumar, Percy Liang, and Tengyu Ma. Verified uncertainty calibration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations (ICLR)*, 2020.

E. L. Lehmann and Joseph P. Romano. Generalizations of the familywise error rate. *Annals of Statistics*, 33(3), 2005.

Jing Lei. Classification with confidence. *Biometrika*, 101 (4):755–769, 10 2014.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.

Andreas Mayr, Günter Klambauer, Thomas Unterthiner, Marvin Steijaert, Joerg Wegner, Hugo Ceulemans, Djork-Arné Clevert, and Sepp Hochreiter. Large-scale comparison of machine learning methods for drug target prediction on chembl. *Chemical Science*, 9(24), 2018.

Allan H. Murphy and Edward S. Epstein. Verification of probabilistic predictions: A brief review. *Journal of Applied Meteorology and Climatology*, 6(5):748 – 755, 1967.

Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, 1996. ISBN 0387947248.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *International Conference on Machine Learning (ICML)*, 2005.

Harris Papadopoulos. Inductive conformal prediction: Theory and application to neural networks. In *Tools in Artificial Intelligence*, chapter 18. IntechOpen, Rijeka, 2008.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

Joseph Romano and Michael Wolf. Control of generalized error rates in multiple testing. *Annals of Statistics*, 35(4), 2007.

Yaniv Romano, Evan Patterson, and Emmanuel Candès. Conformalized quantile regression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Yaniv Romano, Matteo Sesia, and Emmanuel J. Candès. Classification with valid and adaptive coverage. In *Advances in Neural Information Processing (NeurIPS)*, 2020.

Mauricio Sadinle, Jing Lei, and Larry Wasserman. Least ambiguous set-valued classifiers with bounded error levels. *Journal of American Statistical Association*, 2019.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research (JMLR)*, 9:371–421, June 2008.

Emil Spjøtvoll. On the Optimality of Some Multiple Comparison Procedures. *The Annals of Mathematical Statistics*, 43(2):398 – 411, 1972.

Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 2003.

Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas Schön. Evaluating model calibration in classification. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

Vladimir Vovk and Ivan Petej. Venn-abers predictors. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2014.

Vladimir Vovk, Glenn Shafer, and Ilia Nouretdinov. Self-calibrating probability forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2004.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer-Verlag, Berlin, Heidelberg, 2005.

Vladimir Vovk, Ivan Petej, and Valentina Fedorova. Large-scale probabilistic predictors with and without guarantees of validity. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Vladimir Vovk, Valentina Fedorova, Ilia Nouretdinov, and Alexander Gammerman. Criteria of efficiency for conformal prediction. In *International Symposium on Conformal and Probabilistic Prediction with Applications - Volume 9653*, 2016.

Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, 2019.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Zexuan Zhong and Danqi Chen. A frustratingly easy approach for entity and relation extraction. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2021.

A Mathematical details

A.1 Proof of Theorem 3.1

Proof. This is a well-known result (Vovk et al., 2005; Papadopoulos, 2008; Lei et al., 2018; Romano et al., 2019); we prove it here for completeness. Since the nonconformity scores V_i are constructed symmetrically, then

$$\begin{aligned}&((X_1, Y_1), \ldots, (X_{n+1}, Y_{n+1})) \stackrel{d}{=} ((X_{\sigma(1)}, Y_{\sigma(1)}), \ldots, (X_{\sigma(n+1)}, Y_{\sigma(n+1)}))\\ &\iff (V_1, \ldots, V_{n+1}) \stackrel{d}{=} (V_{\sigma(1)}, \ldots, V_{\sigma(n+1)})\end{aligned}$$

for all permutations $(\sigma(1), \ldots \sigma(n+1))$. Therefore, if $\{(X_i, Y_i)\}_{i=1}^{n+1}$ are exchangeable, then so too are their nonconformal scores $\{V_i = \mathcal{M}(X_i, Y_i)\}_{i=1}^{n+1}$ given exchangeability-preserving nonconformity measure $\mathcal{M}$.

By the construction of $\mathcal{C}$, we have

$$Y_{n+1} \in \mathcal{C}_k(X_{n+1}) \iff V_{n+1} \leq \text{Quantile}(1-\epsilon, V_{1:n} \cup \{\infty\}).$$

This implies that V_{n+1} is ranked among the $\lceil (1-\epsilon)\cdot(n+1) \rceil$ smallest of $V_1, \ldots, V_n, \infty$. Since V_i are exchangeable, this happens with probability at least $1-\epsilon$. □

A.2 Proof of Lemma 4.3

Proof. We will not rely on nestedness of $\mathcal{S}_j$.

Notice that

$$t \leq t' \implies \{v_j \colon v_j < t\} \subseteq \{v_j \colon v_j < t'\}. \tag{A.1}$$

As an immediate consequence,

$$\begin{aligned}t \leq t' &\implies \{\text{FP}(z, \mathcal{S}_j) \colon v_j \leq t\} \subseteq \{\text{FP}(z, \mathcal{S}_j) \colon v_j \leq t\} && \text{(A.2)}\\ &\implies \max\{\text{FP}(z, \mathcal{S}_j) \colon v_j < t\} \leq \max\{\text{FP}(z, \mathcal{S}_j) \colon v_j < t'\} && \text{(A.3)}\\ &\implies \text{FP}_{\max}(x, z, t) \leq \text{FP}_{\max}(x, z, t'). && \text{(A.4)}\end{aligned}$$

□

A.3 Proof of Theorem 4.4

Our proof of Theorem 4.4 builds on marginal RCPS (Angelopoulos et al., 2021b). We restate their results here:

Theorem A.1 (Marginal RCPS)**.** *Let $L_i \colon \mathbb{R} \to \mathbb{R}$, $i = 1, \ldots, n+1$ be exchangeable functions, where $L_i(t)$ is non-increasing in t. Also, take $g \colon \mathbb{R} \to \mathbb{R}$ where $g(x)$ is non-decreasing in x. Further assume that $g \circ L_i$ is right-continuous, and*

$$\inf_t g(L_i(t)) < \gamma, \qquad \sup_t g(L_i(t)) \leq B < \infty \text{ almost surely}. \tag{A.5}$$

For any $\gamma \geq 0$, define the random variable $T(\gamma, g)$ as

$$T(\gamma; g) := \inf\left\{ t \colon \frac{1}{n+1}\sum_{i=1}^{n} g(L_i(t)) \leq \gamma \right\}. \tag{A.6}$$

Then $\mathbb{E}[g \circ L_{n+1}(T(\gamma; g))] \leq \gamma + \frac{B}{n+1}$.

Proof. See Angelopoulos et al. (2021b). □

We also restate Corollary 1 of Angelopoulos et al. (2021b).

Corollary A.2 (Marginal RCPS, adjusted)**.** *Under the same setting as in Theorem A.1,*

$$\mathbb{E}[g \circ L_{n+1}(\widetilde{T}(\gamma; g))] \leq \gamma, \tag{A.7}$$

where

$$\widetilde{T}(\gamma; g) = \inf \left\{ t \colon \frac{1}{n+1} \left(B + \sum_{i=1}^{n} g(L_i(t)) \right) \leq \gamma \right\}. \tag{A.8}$$

Proof. See Angelopoulos et al. (2021b). □

Following their analysis, we provide an additional corollary for lower-bounding function R_{n+1}, where $R_1, \ldots, R_{n+1}$ are now non-decreasing exchangeable functions (as opposed to the non-increasing).

Corollary A.3 (Marginal RCPS, lower bound, non-decreasing case)**.** *Similar to the setting in Theorem A.1, let $R_i \colon \mathbb{R} \to \mathbb{R}$, $i = 1, \ldots, n+1$ be exchangeable functions, where $R_i(t)$ is non-decreasing in t. Also, take $g \colon \mathbb{R} \to \mathbb{R}$ where $g(x)$ is non-decreasing in x. Further assume that $g \circ R_i$ is right-continuous, and*

$$\inf_t g(R_i(t)) \geq 0, \qquad \sup_t g(R_i(t)) > C \geq \gamma \text{ almost surely}. \tag{A.9}$$

For any $\gamma \leq 0$, define the random variable $T(\gamma, g)$ as

$$T(\gamma; g) := \inf \left\{ t \colon \frac{1}{n+1} \sum_{i=1}^{n} g(R_i(t)) \geq \gamma \right\}, \tag{A.10}$$

where we define $\inf \varnothing = \infty$. *Then* $\mathbb{E}[g \circ R_{n+1}(T(\gamma; g))] \geq \gamma$.

Proof. Let

$$T'(\gamma; g) := \inf \left\{ t \colon \frac{1}{n+1} \sum_{i=1}^{n+1} g(R_i(t)) \geq \gamma \right\}. \tag{A.11}$$

Since $\inf_t g(R_i(t)) \geq 0$, $\sup_t g(R_i(t)) > C \geq \gamma$, $T'(\gamma; g)$ and $T(\gamma, g)$ are both well-defined almost surely.

Since $\inf_t g(R_i(t)) \geq 0$,

$$\frac{1}{n+1} \sum_{i=1}^{n} g(R_i(t)) \geq \gamma \implies \frac{1}{n+1} \sum_{i=1}^{n+1} g(R_i(t)) \geq \gamma. \tag{A.12}$$

Thus, $T'(\gamma; g) \leq T(\gamma; g)$. Since $g(R_i(t))$ is non-decreasing in t,

$$\mathbb{E}[g \circ R_{n+1}(T(\gamma; g))] \geq \mathbb{E}[g \circ R_{n+1}(T'(\gamma; g))]. \tag{A.13}$$

Let E_f be the unordered set (bag) of $\{R_1, \ldots, R_{n+1}\}$. Then $T'(\gamma; g)$ is a function of E_f, and is a constant conditional on E_f. Exchangeability of R_i and right-continuity of $g \circ R_i$ imply

$$\mathbb{E}[g \circ R_{n+1}(T'(\gamma; g)) \mid E_f] = \frac{1}{n+1} \sum_{i=1}^{n+1} g \circ R_i(T'(\gamma; g)) \geq \gamma. \tag{A.14}$$

As this is true given any E_f, we can take the expectation over E_f to yield

$$\mathbb{E}[\mathbb{E}[g \circ R_{n+1}(T'(\gamma; g)) \mid E_f]] \geq \gamma. \tag{A.15}$$

The proof is completed by applying Eq. (A.13). □

We now prove Theorem 4.4.

Proof. By Lemma 4.3, we have that $\mathrm{FP}_{\max}(x, z, t)$ is non-decreasing in t. It is also easy to verify that $\mathrm{FP}_{\max}(x, z, t)$ is left-continuous in t and preserves exchangeability, so that $\mathrm{FP}_{\max}(X_i, Z_i, t)$ are exchangeable functions of t. Next, define

$$\mathrm{FP}^{-}_{\max}(x, z, t) := \mathrm{FP}_{\max}(x, z, -t), \tag{A.16}$$

so that $\mathrm{FP}^{-}_{\max}(x, z, t)$ is non-increasing in t and right-continuous. Define random variables T'_k and $T'_{k,\delta}$ as

$$T'_k = \inf\left\{t \in \mathbb{R}\colon \frac{B + \sum_{i=1}^{n} \mathrm{FP}^{-}_{\max}(X_i, Z_i, t)}{n+1} \le k\right\} \quad \text{and} \tag{A.17}$$

$$T'_{k,\delta} = \inf\left\{t \in \mathbb{R}\colon \frac{\sum_{i=1}^{n} \mathbf{1}\{\mathrm{FP}^{-}_{\max}(X_i, Z_i, t) \le k\}}{n+1} \ge 1 - \delta\right\}. \tag{A.18}$$

We then have $T_k = -T'_k$ and $T_{k,\delta} = -T'_{k,\delta}$, which gives

$$\mathbb{E}\Big[\mathrm{FP}_{\max}(X_{n+1}, Z_{n+1}, T_k)\Big] = \mathbb{E}\Big[\mathrm{FP}^{-}_{\max}(X_{n+1}, Z_{n+1}, T'_k)\Big] \quad \text{and} \tag{A.19}$$

$$\mathbb{P}\Big(\mathrm{FP}_{\max}(X_{n+1}, Z_{n+1}, T_{k,\delta}) \le k\Big) = \mathbb{P}\Big(\mathrm{FP}^{-}_{\max}(X_{n+1}, Z_{n+1}, T'_{k,\delta}) \le k\Big). \tag{A.20}$$

(Part 1) We first prove $\mathbb{E}\big[\mathrm{FP}^{-}_{\max}(X_{n+1}, Z_{n+1}, T'_k)\big] \le k$.

Since B is finite, we have that $\sup_t \mathrm{FP}^{-}_{\max}(x, z, t) \le \max_j |\mathcal{S}_j| \le B < \infty$. As we assume nonconformity scores are finite, we also have $\inf_t \mathrm{FP}^{-}_{\max}(x, z, t) = 0 < k \in \mathbb{R}_{>0}$. Let $L_i(t) = \mathrm{FP}^{-}_{\max}(X_i, Z_i, t)$ and $g(x) = x$. Corollary A.2 gives

$$\mathbb{E}\Big[\mathrm{FP}^{-}_{\max}(X_{n+1}, Z_{n+1}, T'_k)\Big] \le k. \tag{A.21}$$

Substituting Eq. A.19 gives $\mathbb{E}\big[\mathrm{FP}_{\max}(X_{n+1}, Z_{n+1}, T_k)\big] \le k$.

(Part 2) We now prove $\mathbb{P}\Big(\mathrm{FP}^{-}_{\max}(X_{n+1}, Z_{n+1}, T'_{k,\delta}) \le k\Big) \ge 1 - \delta$.

Let $L_i(t) = \mathbf{1}\{\mathrm{FP}^{-}_{\max}(X_i, Z_i, t) \le k\}$. Let $g(x) = x$. As shown earlier, $\mathrm{FP}^{-}_{\max}(X_i, Z_i, t)$ is non-increasing, right-continuous; as a result $L_i(t)$ is non-decreasing, right-continuous. Let $\gamma = 1 - \delta \in (0, 1)$. Since $g(L_i(t)) \in \{0, 1\}$ and $V_{i,j}$ are finite, it is easy to see that we have $\sup_t g(L_i(t)) = 1 \ge \gamma$ and $\inf_t g(L_i(t)) = 0 \ge 0$.

Applying Corollary A.3 gives

$$\mathbb{E}\big[\mathbf{1}\{\mathrm{FP}^{-}_{\max}(X_{n+1}, Z_{n+1}, T'_{k,\delta}) \le k\}\big] = \mathbb{P}\Big(\mathrm{FP}^{-}_{\max}(X_{n+1}, Z_{n+1}, T'_{k,\delta}) \le k\Big) \tag{A.22}$$

$$\ge \gamma \tag{A.23}$$

$$= 1 - \delta. \tag{A.24}$$

Substituting Eq. A.20 gives $\mathbb{P}\Big(\mathrm{FP}_{\max}(X_{n+1}, Z_{n+1}, T_{k,\delta}) \le k\Big) \ge 1 - \delta$.

□

A.4 Proof of Proposition 4.8

Proof. We first prove simultaneous validity of candidate sets indexed by $j \in \mathcal{J}$. By definition we have

$$\mathrm{FP}(Z_{n+1}, \mathcal{S}_j) \le \mathrm{FP}_{\max}(X_{n+1}, Z_{n+1}, T_\circ) \quad \forall j \in \mathcal{J}, \tag{A.25}$$

which implies

$$\mathbb{E}\Big[\mathrm{FP}(Z_{n+1}, \mathcal{S}_j)\Big] \le \mathbb{E}\Big[\mathrm{FP}_{\max}(X_{n+1}, Z_{n+1}, T_k)\Big] \quad \text{and} \tag{A.26}$$

$$\mathbb{P}\Big(\mathrm{FP}(Z_{n+1}, \mathcal{S}_j) \le k\Big) \ge \mathbb{P}\Big(\mathrm{FP}_{\max}(X_{n+1}, Z_{n+1}, T_{k,\delta}) \le k\Big) \tag{A.27}$$

simultaneously $\forall j \in \mathcal{J}$. Theorem 4.4 then implies validity.

We now show maximal TPR (a simple outcome). If $\mathcal{S}_j \subseteq \mathcal{S}_{j'}$ then $y_c \in \mathcal{S}_j \implies y_c \in \mathcal{S}_{j'}$ for any $y_c \in z \subseteq \mathcal{Y}$. Therefore

$$\mathcal{S}_j \subseteq \mathcal{S}_{j'} \implies \text{TPP}(z, \mathcal{S}_j) \leq \text{TPP}(z, \mathcal{S}_{j'}). \tag{A.28}$$

Since candidate sets are nested,

$$j \leq j' \implies \mathcal{S}_j \subseteq \mathcal{S}_{j'}, \tag{A.29}$$

and

$$\text{TPP}(z, \mathcal{S}_{\max \mathcal{J}}) \geq \text{TPP}(z, \mathcal{S}_{j'}) \quad \forall j' \in \mathcal{J}. \tag{A.30}$$

Since this is true for all (x, z),

$$\mathbb{E}\big[\text{TPP}(Z_{n+1}, \mathcal{S}_{\max \mathcal{J}})\big] = \sup_{h \in \mathcal{H}} \mathbb{E}\big[\text{TPP}(Z_{n+1}, \mathcal{S}_{h \circ \mathcal{J}})\big] \tag{A.31}$$

where $\mathcal{H}$ is the space of all possible index selection policies. □

B Additional training details

In this section, we provide additional details on how training data is constructed to train $\mathcal{F}$.

First, recall that in our setup we have an input space $\mathcal{X}$ and label space $\mathcal{Y}$. For every input $x \in \mathcal{X}$, there are (assumed to be) potentially multiple true positives $z \subseteq \mathcal{Y}$. Referring to Algorithm 1, during training (TRAIN) we split whatever multi-label training data we have into two sets. The first (larger split) is used to learn a likelihood model (which can either be simply training independent binary predictors, or something with label dependencies that are explicitly accounted for, e.g., with a CRF). The second (smaller split) is used to train $\mathcal{F}$. From this second split of data, again, we have inputs x and sets of true positives z. For every input x, however, we can sample combinatorially many candidate output sets $z' \subseteq \mathcal{Y}$, and measure the (ground truth) number of false positives by comparing z' with z. This (x, z') candidate is then used as a training instance for $\mathcal{F}$, where our target is to directly estimate the number of false positives, $|z \setminus z'|$.

In practice, we construct candidate sets greedily by ranking output labels individually, and then combining them into nested sets, i.e., $\{y_{\pi(1)}\}, \{y_{\pi(1)}, y_{\pi(2)}\}, \{y_{\pi(1)}, y_{\pi(2)}, y_{\pi(3)}\}$, etc. This is aligned with our eventual calibration and inference time procedure. As with testing, we only train on candidate sequences up to length B (recall that B is used as a hyper-parameter for a cutoff for the maximum number of considered sets—both for efficiency and technical details for ensuring CP guarantees). This also allows us to train $\mathcal{F}$ quite efficiently, since, instead of randomly sampling candidate sets z', we create them greedily (as we would during testing). Concretely, suppose for an input x with true label set z we have the set of labels ranked by individual likelihood, $\{y_{\pi(1)}, y_{\pi(2)}, y_{\pi(3)}, \ldots, y_{\pi(B)}\}$. For each ranked $y_{\pi(i)}$, we also have a corresponding $\{0, 1\}$ target $y'_{\pi(i)}$ that indicates if $y_{\pi(i)}$ is a false positive, or not. Suppose, in this example, this sequence of labels is $\{1, 0, 1, \ldots, 1\}$. Taking the cumulative sum, our total number of FP targets for the batch are then then provided by the set $\{1, 1, 2, \ldots, B - |z|\}$. We train $\mathcal{F}$ to predict each of these targets by pairing the shared input x with each of the nested candidate sets $\{y_{\pi(1)}, \ldots, y_{\pi(k)}\} \subseteq \{y_{\pi(1)}, \ldots, y_{\pi(B)}\}$, paired with the sum of its false positives, $\sum_{i \leq k} y'_{\pi(i)}$. In total, for a training split of n examples, we will have $n \times B$ prediction targets.

C Implementation and dataset details

In-silico screening. We construct a molecular property screening task using the ChEMBL dataset (see Mayr et al., 2018). Given a specified constraint such as "*is active for property A but not property B*," we want to retrieve at least one molecule from a given set of candidates that satisfies this constraint. The input for each molecule is its SMILES string, a notational format that specifies its molecular structure. The motivation of this task is to simulate in-silico screening for drug discovery, where it is often the case where chemists are searching for a new molecule that satisfies several constraints (such as toxicity and efficacy limits), out of a pool of many possible candidates.

We split the ChEMBL dataset into a 60-20-20 split of molecules, where 60% of molecules are separated into a train set, 20% into a validation set, and 20% into a test set. Next, we take all properties that have at least 50 positive and negative examples (to avoid highly imbalanced properties). Of these properties, we take all N choose K combinations that have at least 100 molecules with all K properties labelled (ChEMBL has many missing values). We set K to 2. For each combination, we randomly sample an assignment for each property (i.e., $\{\text{active}, \text{inactive}\}^K$). We discard combinations for which more than

90% of labeled molecules satisfy the constraint. We keep 5000 combinations for the test set, 767 for validation, and 4375 for training. The molecules for each of the combinations are only sourced from their respective splits (i.e., molecular candidates for constraints in the property combination validation split only come from the molecule validation split). Therefore, at inference time, given a combination we have never seen before, on a molecules we have never seen before, we must try to retrieve the molecules that have the desired combination assignment.

Our directed Message Passing Neural Network (MPNN) is implemented using the `chemprop`[3] repository (Yang et al., 2019). The MPNN model uses graph convolutions to learn a deep molecular representation, that is shared across property predictions. Each property value (active/inactive) is predicted using an independent classifier head. The final prediction is based on an ensemble of 5 models, trained with different random seeds. Given a combination assignment $(Z_1 = z_1, \ldots, Z_k = z_k)$, we naively compute the joint likelihood independently, i.e.,

$$p_\theta(Z_1 = z_1, \ldots, Z_k = z_k) = \prod p_\theta(Z_i = z_i). \tag{C.1}$$

Object detection. As discussed in §5, we use the MS-COCO dataset (Lin et al., 2014) to evaluate our conformal object detection. MS-COCO consists of images of complex everyday scenes containing 80 object categories (such as person, bicycle, dog, car, etc.), multiple of which may be contained in any given example. Since the official test set is hidden, we use the $5k$ examples from the development set and randomly partition them into sets of size $1k$, $1k$, and $3k$ for calibration, validation, and testing, respectively. The EfficientDet model (Tan et al., 2020)[4] for extracting bounding boxes uses a pipeline of three neural networks to extract deep features, and then predict candidates. The model also uses a non-maximum suppression (NMS) post-processing step to reduce the total number of predictions by keeping only the one with the maximum score across highly overlapping prediction boxes. We merge the predictions of all classes into a unified set, where each element is a tuple of (class, bounding box). This means that multiple class predictions can be included for the same bounding box (i.e., there is class uncertainty), and multiple bounding boxes can be found for the same class (i.e., there are multiple objects in one image). We define true positives as predictions that have an intersection over union (IoU) value > 0.5 with a gold bounding box annotation, *and* that match the annotation's class.

Entity extraction. Entity extraction is a popular task in natural language processing. Given a sentence, such as "*Barack Obama was born in Hawaii*," the goal is to identify and classify all named entities that appear—i.e., ("Barack Obama", Person) and ("Hawaii", Location). We use the CoNLL NER dataset (Tjong Kim Sang and De Meulder, 2003), and extract $1k$ examples for calibration out of the $3.3k$ development set, and report test results on the $3.5k$ test set. For our base model, we use the entity extraction module of PURE (Zhong and Chen, 2021), that predicts span scores with a classifier head on top of Albert-base (Lan et al., 2020) contextual embeddings. The classification head has two non-linear layers and uses the learned contextual embeddings of the span start and end tokens, concatenated with a learned span width embedding. We train the model on the training set of the CoNLL NER dataset. We use the official code repository[5] and the following parameters: $1e-5$ learning rate, $5e-4$ task learning rate, 32 train batch size, and 100 context window. Similar to our object detection task, we treat exact span predictions of the correct category as true positives, and any other entity predictions as false positives. As illustrated in Table 1, a fairly large number of sentences do not contain any entities at all, while other sentences may contain several.

C.1 Definition of size-stratified false positive violation

The size-stratified false positive (SSFP) violation measures the worst-case violation of our metric of interest (i.e., expectation or probability), conditioned on the *size* of the output set $\mathcal{C}$. Specifically, SSFP_k and $\text{SSFP}_{k,\delta}$ are defined as follows:

$$\text{SSFP}_k(\mathcal{C}, \{\mathcal{A}\}_{s=1}^a) := \sup_s \max\left(\widehat{\mathbb{E}}\Big[\text{FP}(Z_{n+1}, \mathcal{C}_k(X_{n+1})) \;\Big|\; \{|\mathcal{C}_k(X_{n+1})| \in \mathcal{A}_s\}\Big] - k, 0\right) \quad \text{and} \tag{C.2}$$

$$\text{SSFP}_{k,\delta}(\mathcal{C}, \{\mathcal{A}\}_{s=1}^a) := \sup_s \max\left(\widehat{\mathbb{E}}\Big[\mathbf{1}\{\text{FP}(Z_{n+1}, \mathcal{C}_k(X_{n+1})) > k\} \;\Big|\; \{|\mathcal{C}_k(X_{n+1})| \in \mathcal{A}_s\}\Big] - \delta, 0\right), \tag{C.3}$$

where $\{\mathcal{A}\}_{s=1}^a$ forms a partition of $\{1, \ldots, |\mathcal{Y}|\}$, and $\widehat{\mathbb{E}}$ denotes the empirical average over our test samples.

Following Angelopoulos et al. (2021c), we show that if conditional validity holds for our objectives, then validity also holds after stratifying by set-size. Poor SSFP is therefore a symptom of poor conditional validity.

[3] https://github.com/chemprop/chemprop

[4] We use tf_efficientdet_d2 from https://github.com/rwightman/efficientdet-pytorch.

[5] https://github.com/princeton-nlp/PURE.

In the following, we drop dependence on $n+1$ for clarity.

Proposition C.1 (Expectation case)**.** *Suppose* $\mathbb{E}[\mathrm{FP}(Z, \mathcal{C}_k(X)) \mid X = x] \leq k$ *for each* $x \in \mathcal{X}$*. Then,*

$$\mathbb{E}\Big[\mathrm{FP}(Z, \mathcal{C}_k(X)) \mid \{|\mathcal{C}_k(X)| \in \mathcal{A}\}\Big] \leq k, \quad \textit{for any } \mathcal{A} \subset \{0, 1, 2, \ldots\}. \tag{C.4}$$

Proof.

$$\mathbb{E}[\mathrm{FP}(Z, \mathcal{C}_k(X)) \mid \{|\mathcal{C}_k(X)| \in \mathcal{A}\}] = \frac{\mathbb{E}[\mathrm{FP}(Z, \mathcal{C}_k(X)) \cdot \mathbf{1}\{|\mathcal{C}_k(X)| \in \mathcal{A}\}]}{\mathbb{P}(|\mathcal{C}_k(X)| \in \mathcal{A})} \tag{C.5}$$

$$= \frac{\mathbb{E}[\mathbb{E}[\mathrm{FP}(Z, \mathcal{C}_k(X)) \cdot \mathbf{1}\{|\mathcal{C}_k(X)| \in \mathcal{A}\} \mid X = x]]}{\mathbb{P}(|\mathcal{C}_k(X)| \in \mathcal{A})} \tag{C.6}$$

$$= \frac{\int_x \mathbb{E}[\mathrm{FP}(Z, \mathcal{C}_k(X)) \mid X = x] \cdot \mathbf{1}\{|\mathcal{C}_k(x)| \in \mathcal{A}\} d\mathbb{P}(x)}{\mathbb{P}(|\mathcal{C}_k(X)| \in \mathcal{A})} \tag{C.7}$$

$$\leq \frac{\int_x k \cdot \mathbf{1}\{|\mathcal{C}_k(x)| \in \mathcal{A}\} d\mathbb{P}(x)}{\mathbb{P}(|\mathcal{C}_k(X)| \in \mathcal{A})} \tag{C.8}$$

$$= k. \tag{C.9}$$

□

Proposition C.2 (Probability case.)**.** *Suppose* $\mathbb{P}(\mathrm{FP}(Z, \mathcal{C}_{k,\delta}(X)) \leq k \mid X = x) \geq 1 - \delta$ *for each* $x \in \mathcal{X}$*. Then,*

$$\mathbb{P}\Big(\mathrm{FP}(Z, \mathcal{C}_{k,\delta}(X)) \leq k \mid \{|\mathcal{C}_{k,\delta}(X)| \in \mathcal{A}\}\Big) \geq 1 - \delta, \quad \textit{for any } \mathcal{A} \subset \{0, 1, 2, \ldots\}. \tag{C.10}$$

Proof.

$$\mathbb{P}(\mathrm{FP}(Z, \mathcal{C}_{k,\delta}(X)) \leq k \mid \{|\mathcal{C}_{k,\delta}(X)| \in \mathcal{A}\}) = \frac{\int_x \mathbb{P}(\mathrm{FP}(Z, \mathcal{C}_{k,\delta}(X)) \leq k \mid X = x) \cdot \mathbf{1}\{|\mathcal{C}_{k,\delta}(x)| \in \mathcal{A}\} d\mathbb{P}(x)}{\mathbb{P}(|\mathcal{C}_{k,\delta}(X)| \in \mathcal{A})} \tag{C.11}$$

$$\geq \frac{\int_x (1 - \delta) \cdot \mathbf{1}\{|\mathcal{C}_{k,\delta}(x)| \in \mathcal{A}\} d\mathbb{P}(x)}{\mathbb{P}(|\mathcal{C}_{k,\delta}(X)| \in \mathcal{A})} \tag{C.12}$$

$$= 1 - \delta. \tag{C.13}$$

□

D Additional experimental details

In this section we provide additional discussion of our experimental results.

D.1 Non-smoothness of SSFP results

The top rows of Figure 3 and Figure 2 plot the *worst-case* size-stratified violation, that is the worst-case exceedance of k, conditioned on the prediction set being a particular size. As can be seen from the plots, this can "spike" at different values of ϵ. We provide an interpretation here. As the false positive allowance k grows, the calibration threshold rises, but not all prediction sets necessarily grow in size at the same rate. Therefore, the "worst" sets experience discrete jumps in total false positives at various increments of k. This is most severe when the number of true positives is naturally very low, as in entity extraction, so that increases in set size often lead to increases in false positives (in the worst case). The exact behavior depends on the score function used. In the meantime, between these jumps, the worst case deviation will decrease, as deviation is measured as $\max(\mathbb{E}[\text{false positives} \mid \text{set size}] - k, 0)$, where k is linearly increasing. The *average* deviation is much smoother, but less interesting to measure for comparing various methods, which is why we focus on SSFP.

E Practical considerations and limitations

In this section we address a number of practical considerations, limitations, and extensions for our FP-CP method.

E.1 Choosing a suitable k

An outstanding question a practitioner faces is how to choose the value of k for k-FP and (k, δ)-FP objectives. The value of k in our method has a reliable and easy interpretation: it is the total number of incorrect answers. For many tasks, such as

in-silico screening, there is a direct relation between the number of noisy predictions (e.g., failed experiments conducted during wet-lab validation) and total "wasted" cost. Therefore, for example, given some approximate budget Q and cost per noisy prediction c, a reasonable approach is to then set $k \approx Q/c$. Of course, the advantage of our approach is that the user may set k to whatever they wish—this might change based on their needs, and is not part of our algorithm.

E.2 Choosing between k-fwer and fdr control

A related question to E.1 is when to target k-FWER (i.e., our k-FP and (k, δ)-FP objectives) or FDR (e.g., using Angelopoulos et al. (2021a)). This choice is well discussed in the multiple testing literature (Lehmann and Romano, 2005; Romano and Wolf, 2007; Gold et al., 2009). An important aspect to consider is the size of the label space $\mathcal{Y}$, natural rate of true and false positives, and the efficiency of the base model at separating true positives from false positives. When the total number of true positives is large and $|\mathcal{Y}|$ is large then it is reasonable to control the FDR. If, however, the natural rate of true positives is low, or they are not well separated from false positives, then the FDR can be high and hard to control. This is especially true for smaller prediction sets (as the ratio of positive to negative labels can be quickly driven down even with the addition of only a few false positives). For illustration, suppose for a given example there is one true positive that is ranked 10th by the base model. For many applications, 10 total predictions (with 9 false positives) is acceptable. Yet, the lowest FDR cutoff that allows for this positive to be discoverable is 0.9 (which, for other examples, may allow for hundreds of false positives—an outcome which is undesirable for some applications, even given a high number of accompanying true positives). To satisfy a lower FDR, the algorithm must output an empty set (with FDR = 0). This remains true even if there are a few (but not many) other true positives: for instance, in the previous example, if predictions 10-20 were also all true positives then the lowest FDR is still only 0.5—specifying a FDR tolerances any lower than this would force an empty set prediction.

E.3 Learning more expressive set functions

Our choice of $\mathrm{DeepSets}$ model is motivated by its property of being a universal approximator for continuous set functions, and by its efficient architecture. Of course, its realized accuracy depends on its exact parameterization and optimization. In terms of input features, in §4.2, we chose a simplistic $\phi(x, y_c)$ for two reasons: (1) we view it's low complexity as an advantage (practitioners can simply plug-in individual multi-label probabilities, or other scalar conformity scores, that most out-of-the-box methods provide into a general framework without having to do any more work for providing additional features), and (2) it is easy to train this light-weight model on smaller amounts of data. Still, this approach can discard potentially helpful information about the input x, and any dependencies between labels y_c and y'_c. For example, if y_c and y'_c are mutually exclusive, then the number of false positives if both are included in $\mathcal{S}$ is at least 1. Using more expressive ϕ that is able to capture and take advantage of this sort of side information about x and y_c is a subject for future work.

E.4 Constructing non-nested candidate sets

We choose to construct nested prediction sets because they are efficient and effective. It is useful to emphasize, however, that nestedness is not necessary for our calibration framework: our procedure still works even when candidate sets are not nested. It only relies on $\mathrm{FP}_{\max}$ remaining monotonic in t, which is preserved even for non-nested candidate sets. That said, generally speaking, considering individual candidates in the order of individual likelihood is a good strategy: this maximizes the expected number of true positives in a set of fixed size. Of course, we are not ranking by the true marginal likelihood, but rather the estimate, $p_\theta(y_c \in Z \mid x)$, and this may introduce some error. In theory, the set function $\mathcal{F}$ may be able to identify higher quality outputs sets $\mathcal{S} \in 2^{\mathcal{Y}}$ by jointly considering all of the included elements (rather than ranking them one-by-one). That said, an unconstrained search process over $2^{\mathcal{Y}}$ is expensive. Furthermore, identifying the final output set with maximal TPR, as we show we do in Proposition 4.8, is no longer trivial. Nevertheless, this is a promising area for future work, can potentially be combined with efficient search or pruning methods (e.g., such as in Fisch et al. (2021a)).