Consistent Accelerated Inference via Confident Adaptive Transformers

Tal Schuster∗ Adam Fisch∗ Tommi Jaakkola Regina Barzilay
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
{tals,fisch,tommi,regina}@csail.mit.edu

Abstract

We develop a novel approach for confidently accelerating inference in the large and expensive multilayer Transformers that are now ubiquitous in natural language processing (NLP). Amortized or approximate computational methods increase efficiency, but can come with unpredictable performance costs. In this work, we present CATs—Confident Adaptive Transformers—in which we simultaneously increase computational efficiency, while guaranteeing a specifiable degree of consistency with the original model with high confidence. Our method trains additional prediction heads on top of intermediate layers, and dynamically decides when to stop allocating computational effort to each input using a meta consistency classifier. To calibrate our early prediction stopping rule, we formulate a unique extension of conformal prediction. We demonstrate the effectiveness of this approach on four classification and regression tasks.

1 Introduction

Large pre-trained language models have become the de facto standard approach for solving natural language processing tasks (Devlin et al., 2019; Liu et al., 2019). Despite their impressive performance, however, their often massive computational burden makes them costly to run (Schwartz et al., 2019; Sharir et al., 2020). Concerns about their efficiency have kindled a large body of research in the field (Sanh et al., 2020; Schwartz et al., 2020; Fan et al., 2020). For multilayered architectures such as the Transformer, a popular approach is adaptive early exiting (Xin et al., 2020b; Schwartz et al., 2020, inter alia). Early exiting takes advantage of the observation that task instances vary in complexity. To do so, "early" classifiers are added on top of the simpler features of intermediate layers in the base model, and can trigger a prediction before the full model is executed. Naively deciding when to preempt computation, however, can result in unpredictable decreases in model accuracy.

Quantifying the uncertainty in a prediction in order to decide when additional computation is needed (or not) is critical to making predictions quickly without excessively sacrificing performance. In this paper, we present Confident Adaptive Transformers (CATs), a general method for increasing Transformer-based model efficiency while remaining confident in the quality of our predictions. Specifically, given a fixed, expensive l-layer model \( F(x) \), we create an amortized model \( G(x) \) that includes early classifiers \( G = \{ F_1, \ldots, F_l \} \) that is provably consistent with the original with arbitrarily high probability (e.g., 95% of the time). This process is illustrated in Figure 1.

Our approach builds on conformal prediction (CP), a model-agnostic and distribution-free frame-
work for creating well-calibrated predictions (Vovk et al., 2005). Concretely, suppose we have been given \(n\) examples, \(X_i \in \mathcal{X}, i = 1, \ldots, n\), as unlabeled calibration data, that have been drawn exchangeably from some underlying distribution \(P\). Let \(X_{n+1} \in \mathcal{X}\) be a new exchangeable test example for which we would like to make a prediction. The aim of our method is to construct \(G\) such that it agrees with \(F\) with distribution-free marginal coverage at a tolerance level \(\epsilon \in (0, 1)\), i.e.,

\[
P(G(X_{n+1}) = F(X_{n+1})) \geq 1 - \epsilon. \tag{1}
\]

We consider \(G\) to be \(\epsilon\)-consistent if the frequency of error, \(G(X_{n+1}) \neq F(X_{n+1})\), does not exceed \(\epsilon\). By design, this ensures that \(G\) preserves at least \((1 - \epsilon)\)-fraction of \(F\)'s original performance. Within these constraints, the remaining challenge is to make \(G\) relatively efficient (e.g., a consistent, but vacuous, model is simply the identity \(G \equiv F\)).

In order to support an efficient \(G\), we need a reliable signal for inferring whether or not the current prediction is likely to be stable or not. Past work (e.g., Schwartz et al., 2020) rely on potentially poorly correlated metrics such as the early classifier’s softmax response. We address this challenge by instead directly learning meta “consistency predictors” for each of the \(l - 1\) early classifiers of our \(l\) layer model, by learning from patterns in past predictions. Figure 2 demonstrates the progression of meta confidence scores across layers when applied to “easy” versus “hard” instances from the VitaminC fact verification task (Schuster et al., 2021).

We pair the scores of our meta classifier for each layer with a stopping rule that is calibrated using a unique twist on standard conformal prediction. Traditionally, CP is used to construct prediction sets that cover the desired target (e.g., \(Y_{n+1}\)) with high probability. We instead invert CP to first generate the multi-label set of inconsistent layers, and then exit at the first layer that falls in its complement. Our algorithm is (1) fast to compute in parallel to the main Transformer, (2) requires only unlabeled data, and (3) is statistically efficient in practice.

We empirically validate our method on four diverse NLP tasks—covering both classification and regression, different label space sizes, and varying amounts of training data. We show that our method constitutes a simple-yet-effective approach to confident adaptive prediction with minimal interventions and desirable theoretical guarantees.

Our main contributions are as follows:

1. A novel theoretical extension of conformal prediction to accommodate adaptive prediction;
2. An effective meta consistency classifier for deriving a confident “early exiting” model;
3. A demonstration of the utility of our framework on both classification and regression tasks, where we provide significant efficiency improvements, while guaranteeing high consistency.
2 Related Work

Adaptive computation. Reducing the computational cost of neural models has received intense interest from the machine learning community. Consequently, many efficiency improvements have been proposed in recent years. A popular approach is model compression—either by distillation (Sanh et al., 2020), in which a smaller model is trained to imitate the larger one, or by model pruning (Fan et al., 2020; Michel et al., 2019), in which superfluous parameters are identified and discarded from the original model. Both approaches, however, result in a single model that is used for all future inputs. Adaptive approaches, on the other hand, adjust the amount of computation needed per example in order to "amortize" the total inference cost (Teerapittayanon et al., 2017; Graves, 2017; Dehghani et al., 2019; Huang et al., 2018; Kaya et al., 2019, etc.). As discussed in §1, our method is inspired by the approach of Schwartz et al. (2020); Xin et al. (2020a), where they preempt computation if the softmax value of any early classifier is above a predefined threshold. Unlike our approach, however, their model is not guaranteed to be accurate—even after softmax calibration (Guo, 2017). Several approaches to early exiting also include fine-tuning stages to improve efficiency (Liu et al., 2020; Geng et al., 2021; Zhou et al., 2020). In this work, we choose to avoid this step in order to allow our model to be widely applied with minimal overhead.

Conformal prediction. CP (Vovk et al., 2005) provides a model-agnostic and finite-sample, distribution-free framework for making predictions with marginal coverage guarantees. CP typically is formulated in terms of prediction sets \( C(X_{n+1}) \), where the coverage guarantee is over the event that \( C \) contains \( Y_{n+1} \). As we discuss in §4, internally our method follows a similar approach in which we try to conservatively identify the inadmissible set of all layers that are inconsistent (and exit at the first layer that falls in that set’s complement). Most pertinent to our work, Cauchois et al. (2020) presents algorithms for efficiently producing conformal multi-label predictions. We leverage similar methods in our model, but formulate our solution in terms of the complement of a multi-label set of inconsistent predictions. Our work also complements several recent directions that explore CP in the context of various risk-mitigating applications (Lei and Candès, 2020; Romano et al., 2020; Bates et al., 2020; Fisch et al., 2021a, inter alia), or meta-learning settings (Fisch et al., 2021b).

3 Early Exiting Transformers

In the following, we describe our dynamic early exiting model. We summarize early classification (following previous work) for convenience (§3.1), and then present our novel meta consistency classifier (§3.2). We focus on classification and regression tasks, where the input \( x \in \mathcal{X} \) is mapped into a series of feature representations by a large model \( \mathcal{F}(x) \), until a prediction \( y \in \mathcal{Y} \) is made. We assume that \( \mathcal{F} \) is a multilayered Transformer (Vaswani et al., 2017) composed of \( l \) layers (although our method can be applied to any multilayer network).

For all of our downstream tasks, we assume that the input is either a sentence \( x_1 \), or a pair of sentences \((x_1, x_2)\). Following standard practice, we take the model input \( x = [\text{CLS}]x_1[\text{SEP}] \), or \( x = [\text{CLS}][\text{SEP}]x_1[\text{SEP}]x_2[\text{SEP}] \). For classification tasks, we use a task-specific head \( \text{softmax}(W_p(\phi(W_h h_{\text{CLS}}))) \), where \( h_{\text{CLS}} \in \mathbb{R}^d \) is the hidden representation of the \([\text{CLS}]\) token, \( \phi \) is a nonlinear activation (e.g., tanh), and \( W_h, W_p \) are linear projections, where \( W_p \in \mathbb{R}^{d \times |\mathcal{Y}|} \) and \( W_h \in \mathbb{R}^{d \times d} \). Regression tasks are treated similarly, but use a 1-d output projection, \( w_o \cdot h_{\text{CLS}} \).

3.1 Early predictors

\( \mathcal{F} \)’s structure yields a sequence of hidden \([\text{CLS}]\) representations, \( \{h^{(1)}_{\text{CLS}}, \ldots, h^{(l)}_{\text{CLS}}\} \), where \( h^{(k)}_{\text{CLS}} \in \mathbb{R}^d \) is the representation after applying layer \( k \). Even without any special fine-tuning, we observe that \( h^{(k)}_{\text{CLS}} \) naturally becomes more informative with respect to the output \( y \) as \( k \to l \). For easy \( x \), many intermediate \( h^{(k)}_{\text{CLS}} \) (\( k \ll l \)), are informative enough to make an accurate prediction.

After each intermediate layer \( k < l \), we train an early classification head that is similar to the head used in \( \mathcal{F} \), but reduce the dimensionality of the first projection to \( W_p^{(k)} \in \mathbb{R}^{d \times d} \). The final \( \mathcal{F}_l \) is unchanged from \( \mathcal{F} \). These extra parameters \((d_e \times d + d_e \times |\mathcal{Y}|)\) are quick to tune on top of a fixed \( \mathcal{F} \), and we can reuse \( \mathcal{F} \)'s training data as \( D_{\text{tune}} \). The classifier \( \mathcal{F}_l(x) = \text{softmax}(W_o^{(k)}(\phi(W^{(k)} h^{(k)}_{\text{CLS}}))) \) is then used after layer \( k \) to make an early prediction candidate. Early regression is handled similarly.

3.2 Meta early exit classifier

To decide \( \text{when} \) to accept the current prediction and stop computation, we require some signal as to how likely it is that \( \mathcal{F}_k(x) = \mathcal{F}(x) \). Previous
work relies on intrinsic measures (e.g., softmax response). Here, we present a new meta classifier to explicitly estimate the consistency of an early predictor. Given fixed $F_k$ and $G$, we train a small binary MLP, $M_k(x) \in \mathbb{R}$, on another unlabeled (limited) sample of task in-domain data, $D_{\text{meta}}$.

As input to $M_k$ for layer $k$, we provide the current “early” hidden state $\phi(W^{(k)}_i h_{[\text{CLST}]}^{(k)}$, in addition to several processed meta features, see Table 1. We then train $M_k$ with a binary cross entropy objective, where we maximize the likelihood of predicting $1\{F_k(x_i) = F(x_i)\}$ for $x_i \in D_{\text{meta}}$.

Using the trained $F_k$ and $M_k$, the full, adaptive model $G$ is then defined by the prediction process

$$G(x) := \begin{cases} F_k(x) & \text{if } R_e(M_k(x)) = 1, \\ F_{k+1}(x) & \text{otherwise}, \end{cases}$$

where $R_e(\cdot) \in \{0,1\}$ is a stopping rule, and $k$ is initialized to $k = 1$. We discuss how to calibrate $R_e(M_k(x))$ such that $G$ satisfies Eq. (1) next.

## 4 Conformalized Early Exits

In order to guarantee $\epsilon$-consistent performance, the stopping rule $R_e(M_k(x))$ has to be well-calibrated. In this section, we present a new CP-based approach.\footnote{See Shafer and Vovk (2008) for a concise review of CP.}

All proofs are deferred to Appendix A.

### 4.1 Problem formulation

Let $I(x) := \{i: F_i(x) \neq F(x)\}$ be the index set of layers that are inconsistent with the final model’s prediction. To maintain $\epsilon$-consistency, we must avoid using any of the predictions specified by this set, $F_i(x)$ where $i \in I(x)$, more than $\epsilon$-fraction of the time. In the following section, we show how meta classifiers $M_{1:1-1}$ can be paired with a conformal procedure $P$ to create $C_\epsilon(x)$, a conservative prediction of $I(x)$, where we ensure that $I(x) \subseteq C_\epsilon(x)$ with probability at least $1 - \epsilon$.

We then take $R_e$ as the simple rule where

$$R_e(M_k(x)) := \begin{cases} 1 & \text{if } k \notin C_\epsilon(x), \\ 0 & \text{otherwise}. \end{cases}$$

Proposition 4.1 states our performance guarantee when $R_e(M_k)$ is paired with $G$ following Eq. (2).

**Proposition 4.1.** Assume that unlabeled examples $X_i$, $i = 1, \ldots, n + 1$ are exchangeable. For any $\epsilon \in (0, 1)$, let $P$ be a conformal procedure that identifies an index set $C_\epsilon$ (based on the first $n$ examples) for test point $X_{n+1}$ such that

$$P(I(X_{n+1}) \subseteq C_\epsilon(X_{n+1})) \geq 1 - \epsilon. \quad (4)$$

Define $K := \min\{j: j \in C_\epsilon^n(X_{n+1})\}$, the layer selected by $G$ following Eqs. (2) and (3). Then

$$P(F_{K}(X_{n+1}) = F(X_{n+1})) \geq 1 - \epsilon. \quad (5)$$

**Remark 4.2.** Note that Eq. (4) is stricter than necessary. Fundamentally, we only require that $P(K \in I^c(X_{n+1})) \geq 1 - \epsilon$. Nevertheless, Eq. (4) is easier to calibrate, and leads to strong empirical results despite being theoretically conservative.

### 4.2 Conformal calibration

Conformal prediction is based on hypothesis testing, where for a given input $x$ and possible output $y$, a statistical test is performed to accept or reject the null hypothesis that the pairing $(x, y)$ is correct. In our setting, we consider the null hypothesis that layer $k$ is inconsistent, and we use $M_k(x)$ as our test statistic. Since $M_k$ is trained to predict $1\{F_k(x_i) = F(x_i)\}$, a small value of $M_k(x)$ indicates how “surprised” we would be if layer $k$ was in fact consistent with layer $l$ for input $x$. Informally, a low level of surprise indicates that the current input “conforms” to past data. To calibrate our test, we use a held-out set of $n$ unlabeled, exchangeable examples, $D_{\text{cal}}$, as described next.

#### 4.2.1 Independent calibration

As a first approach, we construct $C_\epsilon(x)$ by composing $l - 1$ separate tests for $F_k(x) \neq F(x)$, each with significance $\epsilon_k$, where $\epsilon_k$ are corrected for multiple testing. Let $v_{k}^{1:1,\infty}$ denote the inflated empirical distribution of inconsistent layer scores,

$$\{M_k(x_i): x_i \in D_{\text{cal}}, F_k(x_i) \neq F(x_i)\} \cup \{\infty\},$$

where $Q(\alpha, v_{k}^{1:1,\infty})$ is its $\alpha$ quantile. We then predict the inconsistent index set at $x$ as

$$C^{\text{ind}}_\epsilon(x) = \left\{k: M_k(x) \leq Q(1 - \epsilon_k, v_{k}^{1:1,\infty})\right\}. \quad (6)$$

\footnote{Here $A^c$ denotes the complement index set $\{i: i \notin A\}$.}
Theorem 4.3. Let \( \tilde{e}_k = \omega_k \cdot \epsilon \), where \( \omega_k \) is a weighted Bonferroni correction, i.e., \( \sum_{k=1}^{l-1} \omega_k = 1 \). Then \( C_{\text{ind}}^{\epsilon}(X_{n+1}) \) is a valid set that satisfies Eq. (4).

Remark 4.4. \( \omega_{1:k} \) can be tuned on a development set \( D_{\text{dev}} \) as long as \( D_{\text{dev}} \) is distinct from \( D_{\text{cal}} \).

4.2.2 Shared calibration

\( C_{\text{ind}}^{\epsilon} \) has the advantage of calibrating each layer independently. As \( l \) grows, however, \( \tilde{e}_k \) will tend to 0 in order to retain validity (as specified by Theorem 4.3). As a result, \( C_{\text{ind}}^{\epsilon} \) will lose statistical efficiency. Following a similar approach to Cauchois et al. (2020) and Fisch et al. (2021a), we compute a new test statistic, \( M_{\text{max}} \), as

\[
M_{\text{max}}(x) = \max_{k \in [l-1]} \{ M_k(x) : F_k(x) \neq \mathcal{F}(x) \}. \tag{7}
\]

We discard ill-defined values when \( M_{\text{max}}(x) = \max \emptyset \). \( M_{\text{max}}(x) \) reflects the worst-case confidence across inconsistent layers for input \( x \) (i.e., where \( M_k(x) \) predicts a high consistency likelihood for layer \( k \) when layer \( k \) is, in fact, inconsistent). This worst-case statistic allows us to keep a constant significance level \( \epsilon \), even as \( l \) grows. Let \( m^{(1:n, \infty)} \) denote the inflated empirical distribution

\[
\{ M_{\text{max}}(x_i) : x_i \in D_{\text{cal}}, \exists k F_k(x_i) \neq \mathcal{F}(x_i) \} \cup \{ \infty \}.
\]

We then predict the index set at \( x \) as

\[
C_{\text{shr}}^{\epsilon}(x) = \left\{ k: M_k(x) \leq Q(1 - \epsilon, m^{(1:n, \infty)}) \right\}. \tag{8}
\]

Theorem 4.5. For any number of layers \( l \in \mathbb{N} \), \( C_{\text{shr}}^{\epsilon}(X_{n+1}) \) is a valid set that satisfies Eq. (4).

4.3 Conditional conformal calibration

Up until this point, we have been concerned with maintaining a marginal guarantee on \( \mathbb{P}(\mathcal{G}(X_{n+1}) = \mathcal{F}(X_{n+1})) \), where the randomness is over calibration points \( X_{1:n} \) and test point \( X_{n+1} \). In reality, however, we typically only care about consistency when \( \mathcal{F}(x) \) is correct, as making an inconsistent prediction \( \mathcal{G}(x) \neq \mathcal{F}(x) \) in this case will necessarily result in an error. If exchangeable labeled calibration data is available, \((X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}, i = 1, \ldots, n \), then we can consider a more refined version of Eq. (1):

\[
\mathbb{P}(\mathcal{G}(X_{n+1}) = \mathcal{F}(X_{n+1}) | \mathcal{F}(X_{n+1}) = Y_{n+1}) \geq 1 - \epsilon. \tag{9}
\]

Note that this criterion still guarantees the same lower bound on \( \mathcal{G} \)'s performance—i.e., \( (1 - \epsilon)\)-
the following methods are not guaranteed to give well-calibrated performance (as our CP-based are).

**Static.** We use the same number of layers for all inputs. We choose the exit layer as the first one that obtains the desired consistency on average on \( D_{\text{cal}} \).

**Softmax threshold.** Following Schwartz et al. (2020), we exit on the first layer where \( p_k^{\text{max}} \geq 1 - \epsilon \), where \( p_k^{\text{max}} \) denotes the maximum softmax response of our early classifier. Softmax values are calibrated using temperature scaling (Guo, 2017) on another held-out data split, \( D_{\text{scale}} \).

**Meta threshold.** Even if perfectly calibrated, \( p_k^{\text{max}} \) from softmax thresholding is not measuring consistency likelihood \( \mathbb{P}(\hat{G}(X) = F(X) \mid X = x) \), rather \( \mathbb{P}(\hat{G}(X) = Y \mid X = x) \). This is equivalent if \( F \) is an oracle, but breaks down when \( F \) is not. We also experiment with thresholding the confidence value of our meta classifier (§3.2) in a similar way (i.e., exiting when it exceeds \( 1 - \epsilon \)).

### 5.3 Evaluation

For each task, we use a proper training, validation, and test set. We use the training set to learn \( F \) and \( G \). We perform model selection on the validation set, and report final numbers on the test set. For all methods, we report the marginalized results over 25 random trials, where in each trial we partition the data into 80\% \( D_{\text{cal}} \) \( (x_{1:n}) \) and 20\% \( D_{\text{test}} \) \( (x_{n+1}) \). In order to compare the aggregate performance of different methods across all tolerance levels, we plot each metric as a function of \( \epsilon \). In all plots, shaded regions show the 16-84\%th percentiles across trials. We report the following metrics:

**Consistency.** We measure the percent of inputs for which the prediction of the CAT model \( G \) is the same as the full Transformer on our test prediction, i.e., \( G(X_{n+1}) = F(X_{n+1}) \). For regression tasks, we count a prediction as consistent if it is within a small margin \( \tau \) from the reference (we use \( \tau = 0.5 \)). As discussed in §1, if \( G \) is \( \epsilon \)-consistent, we can also derive an average performance lower bound: it will be at least \((1 - \epsilon) \times F's \) average performance.\(^7\)

**Layers (i).** We report the computational cost of the model as the average number of Transformer layers used. Our goal is to improve the efficiency (i.e., use fewer layers) while preserving \( \epsilon \)-consistency. We choose this metric over absolute run-time to allow for implementation-invariant comparisons, but we provide a reference analysis next to permit easy approximate conversions.

### 5.4 Absolute runtime analysis

The exact run-time of \( G \) depends on the efficiency of the hardware, software, and implementation used. Ideally, the early and meta classifiers can run in parallel with the following Transformer layer (layer \( k + 1 \)). As long as they are faster to compute concurrently than a single layer, this will avoid incurring any additional time cost. A naive asynchronous implementation, however, could lead to inefficiencies when using a small tolerance \( \epsilon \).

We provide a reference timing for the IMDB task implemented with the Transformers (Wolf et al., 2020) library, PyTorch 1.8.1 (Paszke et al., 2019), and an A100-PCIE-40GB Nvidia GPU with CUDA 11.2. A full forward path of an Albert-xlarge takes 22.32ms per input, 0.85ms \( \times 24 \) for the transformer layers and 1.95ms for the embedding layer and top classifier. Our early classifier takes 0.20ms and the meta classifier takes 0.11ms. Therefore, with a naive implementation, a CAT model \( G \) with an average exit layer less than 17.6 with the meta classifier, or 19.5 without, will realize an overall reduction in wall-clock time relative to the full \( F \).

### 6 Experimental Results

In the following, we present our main results. We experiment with both our trained meta classifier \( M_k \) confidence score (Meta, §3.2), and, for classification tasks, we also explore using the early classifier’s softmax response, \( p_k^{\text{max}} \) (SM), as a drop-in replacement for \( M_k \) (at no additional computational cost). In Appendix C, we report results with other drop-in \( M_k \) replacements, in addition to results with conditional calibration (§4.3). Appendix E provides qualitative early exit examples.

#### 6.1 Classification results

Figure 3 summarizes the average consistency and number of layers used by \( G \) as a function of \( \epsilon \), while Table 3 presents results for specific \( \epsilon \) on task test sets. Independent calibration proves to be quite conservative due to the lose of statistical power from the loose union bound of the Bonferroni correction for large \( l \) (here, \( l = 24 \)). At some levels of \( \epsilon \), non-CP baselines perform competitively, however, they lack formal guarantees. Overall, for the most critical tolerance levels (small \( \epsilon \), right-hand side of the plots), our shared method leads to significant efficiency gains (where better means using fewer layers) while still maintaining the desired...
The effectiveness of our meta predictor, $\mathcal{M}_k$, is most pronounced for tasks with $|Y| > 2$, where the drop-in softmax score (SM) becomes less indicative of consistency. Both SM and Meta are relatively well-calibrated for IMDB and VitaminC, which makes the threshold-based exit rule a competitive baseline. Still, combining our meta confidence predictor with our shared CP-based method provides both reliable and significant gains.

The computational advantage of our CAT model is dependent on the average difficulty of the task and the implementation. As Table 3 shows, allowing up to an $\epsilon$ of 10% inconsistency, for two of the tasks we cut down the average Transformer layer to only 9 out of 24 using our Shared/ Meta model. Following the analysis in §5.4, this leads to an approximate speedup of $1.8 \times$ with an asynchronous implementation and of $2.7 \times$ with a concurrent one, compared to running the full model. Moreover, Figure 4 illustrates the user’s control over available computational resources via modulating $\epsilon$. Decreasing $\epsilon$ increases the confidence level required before committing to the early clas-
Figure 4: Distribution of exit layers per tolerance level $\epsilon$ for the IMDB task (dev set) with Shared/ Meta. Larger $\epsilon$ allows the CAT model to shift its predictions earlier by permitting for more inconsistencies with the full model $F$.

Figure 5: Regression results for STSB (dev).

6.2 Regression results

Table 4 and Figure 5 present results for our regression task, where we see similar trends. Here, an attractive advantage of our meta confidence predictor is its generalizability to multiple task output types. Notice that the event space of $1\{G(X) = F(X)\} = \{0, 1\}$ always, regardless of the original $Y$. This allows it to be easily adapted to tasks beyond classification, such as regression, where traditional softmax-based confidence measures (as used in, e.g., Schwartz et al. (2020)) are absent.

7 Conclusion

The ability to make predictions quickly without excessively degrading performance is critical to production-level machine learning systems. In fact, being capable of quantifying the uncertainty in a prediction and deciding when additional computation is needed (or not) is a key challenge for any intelligent system (e.g., see the System 1 vs. System 2 dichotomy explored in Kahneman (2011)).

In this work, we addressed the challenge of deciding when to sufficiently trust an early prediction of Transformer-based multilayer models by learning from their past predictions. Our Confident Adaptive Transformers (CATs) framework leverages meta predictors to accurately assess whether or not the prediction of a simple, early classifier trained on an intermediate Transformer representation is likely to already be consistent with that of the full model $F(X)$ (i.e., after all $l$ layers of $F$ are computed). Importantly, we develop a new conformal prediction approach for calibrating the confidence of the meta classifier that is (1) fast to compute in lockstep with the Transformer, (2) requires only unlabeled data, and (3) provides statistically efficient marginal guarantees on the event that the prediction of the faster, amortized CAT model is consistent with that of the full $F$. Our results on multiple tasks demonstrate the generality of our approach, and highlight its effectiveness in consistently improving computational efficiency—all while maintaining a reliable margin of error.

Table 4: Test results for the STSB regression task.

| Method       | Consist. | Layers |
|--------------|----------|--------|
| $1 - \epsilon = 0.95$ |          |        |
| Static       | 100.00   | 24.00  |
| Thres./ Meta | 99.87    | 19.19  |
| Indep./ Meta | 99.29    | 23.60  |
| Shared/ Meta | 96.42    |        |
| $1 - \epsilon = 0.90$ |          |        |
| Static       | 92.51    | 20.00  |
| Thres./ Meta | 99.19    | 18.53  |
| Indep./ Meta | 97.77    | 20.26  |
| Shared/ Meta | 92.65    | 17.29  |

8 As long as equality is suitably defined, e.g., for STSB we define consistent outputs as being within $\tau = 0.5$ away.
Acknowledgements

We thank Jiaming Luo and the MIT NLP group for helpful discussions. TS is supported in part by DSO grant DSOCL18002. AF is supported in part by an NSF Graduate Research Fellowship.

References

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

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, MateuszLitwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Maxime Cauchois, Suyash Gupta, and John Duchi. 2020. Knowing what you know: valid confidence sets in multiclass and multilabel prediction. arXiv preprint: arXiv 2004.10181.

Daniel Cer, Mona Diah, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. 2019. Universal transformers. In International Conference on Learning Representations.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Angela Fan, Edouard Grave, and Armand Joulin. 2020. Reducing transformer depth on demand with structured dropout. In International Conference on Learning Representations.

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

Adam Fisch, Tal Schuster, Tommi Jaakkola, and Regina Barzilay. 2021b. Few-shot conformal prediction with auxiliary tasks. arXiv preprint: arXiv 2102.08898.

Shijie Geng, Peng Gao, Zuohui Fu, and Yongfeng Zhang. 2021. Romebert: Robust training of multi-exit bert.

Alex Graves. 2017. Adaptive computation time for recurrent neural networks.

Antonio Gulli. 2004. Ag’s corpus of news articles.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, pages 1321–1330, International Convention Centre, Sydney, Australia. PMLR.

Hongyu Guo. 2017. A deep network with visual text composition behavior. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 372–377, Vancouver, Canada. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition.

Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Weinberger. 2018. Multi-scale dense networks for resource efficient image classification. In International Conference on Learning Representations.

Daniel Kahneman. 2011. Thinking, fast and slow. Farrar, Straus and Giroux, New York.

Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. 2019. Shallow-deep networks: Understanding and mitigating network overthinking. In Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 3301–3310. PMLR.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

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

Lihua Lei and Emmanuel Candès. 2020. Conformal inference of counterfactuals and individual treatment effects. arXiv preprint: arXiv 2006.06138.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. 2020. FastBERT: a self-distilling BERT with adaptive inference time. In
Furthermore, assume that $V_n \leq \alpha$ exchangeable. Then for any $\{V_n\}_{n \geq 1}$ denote the empirical quantiles, we have

$$Q(\alpha; V_n) = \left\{ \frac{i}{n+1} \mid i \in \{0, 1, \ldots, n+1\} \right\}$$

Let $\alpha_n = \frac{n+1}{n} \alpha$. Then for any $\{\alpha_n\}_{n \geq 1}$ denote the empirical quantiles, we have

$$Q(\alpha_n; V_n) = \left\{ \frac{i}{n+1} \mid i \in \{0, 1, \ldots, n+1\} \right\}$$

$\mathbb{P}(V_{n+1} \leq Q(\alpha, V_{1:n} \cup \{\infty\})) \geq \alpha$. 

## Proofs

We first state the following useful lemma on inflated sample quantiles.

**Lemma A.1.** Let $Q(\alpha; F)$ denote the $\alpha$ quantile of distribution $F$. Let $V_{1:n}$ denote the empirical distribution over random variables $\{V_1, \ldots, V_n\}$. Furthermore, assume that $V_i$, $i = 1, \ldots, n + 1$ are exchangeable. Then for any $\alpha \in (0, 1)$, we have

$$\mathbb{P}(V_{n+1} \leq Q(\alpha, V_{1:n} \cup \{\infty\})) \geq \alpha.$$
Accordingly, for the exchangeable $V_{q_k}$ over all probability at least $1$ changeable, then ing exchangeability. Therefore, as $X_{k}$ test point. For all $X_{k}$, and $I = \{1, \ldots, n\}$. By exchangeability, this event occurs with probability at least $\frac{\alpha(n+1)}{n+1} \geq \alpha$.

### A.1 Proof of Proposition 4.1

**Proof.** We prove by simple calculation using the property assumed in Eq. (4).

\[
P(F_K(X_{n+1}) = F(X_{n+1}))
= P(\min_i C^c_i(X_{n+1}) \in \mathcal{I}^c(X_{n+1}))
\geq P(C^c_i(X_{n+1}) \in \mathcal{I}^c(X_{n+1}))
= P(\mathcal{I}(X_{n+1}) \subseteq C_i^{\text{ind}}(X_{n+1}))
\geq 1 - \epsilon.
\]

### A.2 Proof of Theorem 4.3

**Proof.** For a given $k$, let $V_k^{(i)} := M_k(X_i)$ denote the random meta confidence values used for calibration, and $V_k^{(n+1)} := M_k(X_{n+1})$ the random test point. For all $k$, $M_k$ is trained and evaluated on separate data ($D_{\text{meta}}$ vs $D_{\text{cal}} \cup D_{\text{test}}$), preserving exchangeability. Therefore, as $X_{1:n+1}$ are exchangeable, then $V_k^{(1:n+1)}$ are also exchangeable.

Layer $k$ is included in $C^{\text{ind}}_i$ iff $V_k^{(n+1)} \leq Q(1 - \tilde{\epsilon}_k, V_k^{(1:n)}) \cup \{\infty\}$. For a given $k$, this happens with probability at least $1 - \tilde{\epsilon}_k$ by Lemma A.1. Taken over all $k \in \mathcal{I}(X_{n+1})$ where $|\mathcal{I}(X_{n+1})|$ is at most $l - 1$ (i.e., all early layers are inconsistent), we have

\[
\mathbb{P}(\mathcal{I}(X_{n+1}) \subseteq C^{\text{ind}}_i(X_{n+1}))
= 1 - \mathbb{P}\left(\bigcup_{k \in \mathcal{I}} \{k \not\in C^{\text{ind}}_i(X_{n+1})\}\right)
\geq 1 - \sum_{k \in \mathcal{I}} \tilde{\epsilon}_k
\geq 1 - \epsilon.
\]

### A.3 Proof of Theorem 4.5

**Proof.** By the same argument as Theorem 4.3, the meta scores $M_{k}(X_{i})$ are exchangeable. Since $M_{\text{max}}$ operates symmetrically across all $X_{i}$, $M^{(i)} = M_{\text{max}}(X_{i})$ are also exchangeable. Let $M^{(n+1)}$ denote the maximum meta score across inconsistent layers for the new test point. By Lemma A.1, this falls below $Q(1 - \epsilon, M^{(1:n)} \cup \{\infty\})$ with probability at least $1 - \epsilon$. Since $M^{(n+1)}$ reflects the maximum meta score, this entails that the meta scores of all other inconsistent layers $k \in \mathcal{I}(X_{n+1})$ for $X_{n+1}$ will be below $Q(1 - \epsilon, M^{(1:n)} \cup \{\infty\})$ if $M^{(n+1)}$ is, and thereby be included in $C^{\text{shr}}_i(X_{n+1})$. This gives the bound in Eq. (4). \hfill \qed

### A.4 Proof of Corollary 4.6

**Proof.** It suffices to show that the filter $A := \{X_{i} : F(X_{i}) = Y_{i}\}$ produces exchangeable samples $X_{j} \in A$. The condition $F(X_{i}) = Y_{i}$ is symmetric across all $X_{i}$, where $F$ is fixed. Therefore the subset $A$ is also exchangeable conditioned on $F(X_{i}) = Y_{i}$. The setting then reduces to a straightforward application of Proposition 4.1. \hfill \qed

### B Implementation Details

We implement our early exit Transformers (§3) on top of the Transformers library (Wolf et al., 2020). We set $d_{e}$ to 32 in our experiments. For each task we fix a pre-trained $F$ and train the early and meta classifiers. We reuse the same training data that was used for $F$ and divide it to 70/10/20% portions for $D_{\text{tune}}, D_{\text{scale}}$ and $D_{\text{meta}}$, respectively. For classification tasks, we add the temperature scaling step (Guo et al., 2017) after the early training to improve the calibration of the softmax. We run the scaling for 100 steps on $D_{\text{scale}}$ using an Adam optimizer (Kingma and Ba, 2017) with a learning rate of $10^{-3}$. For the early and meta training we use the same optimizer as for $F$. 

11
We fix $\mathcal{F}$ rather than train it jointly with the new components of $\mathcal{G}$ to avoid any reduction in $\mathcal{F}$’s performance (Xin et al., 2020b). This also makes our method simple to train over any existing Transformer without having to retrain the whole model which could be very costly. Training all parameters of $\mathcal{G}$ jointly can lead to more efficient inference as the early representations will be better suited for classification (Schwartz et al., 2020; Geng et al., 2021), but potentially with the cost of reducing the accuracy of $\mathcal{F}$. In the case of joint training, our CATs will provide consistency guarantees with respect to the jointly-trained $\mathcal{F}_l$.

We implement the conformal calibration process in Python and perform retrospective analysis with different random splits of $D_{\text{cal}}$ and $D_{\text{test}}$. For Theorem 4.3, we simply use the uniform Bonferroni correction, setting $w_k = \frac{1}{l-1} \forall k$.

C Additional Results

We provide complementary results for the experiments in the main paper.

C.1 Conditional calibration

As described in §4.3, if $D_{\text{cal}}$ is labeled, we might want to modify the CP criterion to account only for the inputs that $\mathcal{F}$ is correct on. Inconsistencies on those inputs will surely lead to incorrect predictions. We evaluate this version on our three classification tasks and report the results in Table C.1. We also report the conditioned consistency: $E[G(X_{n+1}) = \mathcal{F}(X_{n+1}) | \mathcal{F}(X_{n+1}) = Y_{n+1}]$.

The conditional calibration allows an additional gain in efficiency while meeting the accuracy goal. For example, the number of layers used for the IMDB task with $\epsilon = 0.05$ decreased from 10.75 to 10.02. Yet, the difference in performance is not substantial, demonstrating the strength of our non-conditional calibration with unlabelled data. The difference between the two versions might become more significant in cases where $\mathcal{F}$ is less accurate, resulting in a greater discrepancy between the conditional and non-conditional distributions.

C.2 Nonconformity measures

The test statistic used for a conformal prediction is typically called a nonconformity measure (i.e., in our work this is $M_k(x)$). We experiment with different nonconformity measures as drop-in replacements for $M_k(x)$, and report the results in Table C.2. The conformal calibration guarantees validity with any measure, even a random one, as long as they retain exchangeability. Good measures are ones that are statistically efficient, and will minimize the number of layers required for prediction at the required confidence level. This is a result of smaller $C_\epsilon$ sets, that tightly cover the inconsistent layers (and hence are more judicious with the complement, $C_c^\epsilon$). To be consistent with previous work where softmax metrics are used (such as Schwartz et al., 2020), we use $p_k^{\text{max}}$ as our non-Meta baseline in the main paper. In some settings, however, $p_k^{\text{diff}}$ performs slightly better.

C.3 Exit layer statistics

Figure C.1 depicts the distribution of exit layers for the different tasks with three reference tolerance levels. Reducing $\epsilon$ requires greater confidence before exiting, resulting in later exits on average. We provide example inputs with their respective exit layer in Appendix E.

D Albert-base Results

Figure D.1 reports the classification and regression results with an Albert-base 12-layers model. The trends are similar to the larger 24-layers version. Again, we see the efficacy of our Shared conformal calibration and the Meta nonconformity scores. For example, the AG News CAT Shared/ Meta model can preserve 95% consistency while using less than 5 Transformer layers on average.

E Example Predictions

Table E.1 reports examples of inputs for different tasks and the number of layers that our Albert-xlarge CAT with $\epsilon = 0.1$ required. These examples suggest that “easier” inputs (e.g., containing cue phrases or having large overlaps in sentence-pair tasks) might require less layers. In contrast, more complicated inputs (e.g., using less common language or requiring numerical analysis) can lead to additional computational effort until the desired confidence is obtained.
Given the availability of labeled training data, conditional calibration allows a refined guarantee of consistency only over the inputs that are predicted correctly by $F$. In return, this leads to improved efficiency as the meta classifier performs better on these inputs. We report both conditional and non-conditional consistency results here and observe the validity of both calibration methods. In the main paper, we use non-conditional calibration for all classification tasks since we assume an unsupervised setting.

### Table C.1: Results on the development sets with the Shared/ Meta model comparing the non-conditional (unlabeled) with the conditional calibration (§4.3).

| Conditional calibration | IMDB | | VitaminC | | AG News |
|-------------------------|------|------|-----------|------|--------|
|                         | Consistency | Acc. | Layers | Consistency | Acc. | Layers | Consistency | Acc. | Layers |
| $1 - \epsilon = 0.9$:   | (88.50) | (85.17) | (89.02) |
| $\times$                | 97.84 | 96.99 | 92.24 | 10.75 | 97.69 | 96.91 | 88.29 | 16.49 | 97.40 | 96.98 | 91.98 | 10.60 |
| $\checkmark$            | 97.28 | 96.31 | 91.87 | 10.02 | 97.08 | 96.14 | 87.85 | 16.12 | 96.98 | 96.56 | 91.62 | 10.37 |
| $1 - \epsilon = 0.9$:   | (84.69) | (84.33) | |
| $\times$                | 95.49 | 94.40 | 90.45 | 8.80 | 94.92 | 93.74 | 86.17 | 15.09 | 94.68 | 94.08 | 89.72 | 8.88 |
| $\checkmark$            | 94.74 | 93.56 | 89.88 | 8.39 | 93.94 | 92.63 | 85.41 | 14.72 | 94.13 | 93.48 | 89.27 | 8.58 |

Table C.1: Results of our Shared model on the classification development sets using different nonconformity measures.

- $p_k^{\text{diff}}$ and $p_k^{\text{max}}$ are defined in Table 1 and $H(p_k)$ is the entropy of softmax outputs.
- Our CP-based Shared method provides the guaranteed consistency with any measure, even random. The benefit, however, of using a better measure is in confidently exiting earlier. Our Meta measure allows the use of least Transformer layers meeting the consistency requirement with enough confidence.

### Table C.2: Results of our Shared model on the classification development sets using different nonconformity measures.

| Nonconformity measure | IMDB | | VitaminC | | AG News |
|-----------------------|------|------|-----------|------|--------|
|                       | Consist. | Acc. | Layers | Consist. | Acc. | Layers | Consist. | Acc. | Layers |
| $1 - \epsilon = 0.95$: | (88.50) | (85.17) | (89.02) |
| Random                | 97.23 | 91.56 | 21.57 | 96.91 | 87.42 | 22.71 | 97.11 | 91.58 | 21.60 |
| $\mathcal{H}(p_k)$   | 97.28 | 92.84 | 12.49 | 96.79 | 88.28 | 17.44 | 97.15 | 92.79 | 14.55 |
| $p_k^{\text{diff}}$  | 97.28 | 92.84 | 12.49 | 96.83 | 88.38 | 17.42 | 96.96 | 92.80 | 12.89 |
| $p_k^{\text{max}}$ (SM) | 97.28 | 92.84 | 12.49 | 96.79 | 88.31 | 17.40 | 97.08 | 92.81 | 13.23 |
| Meta                  | 96.99 | 92.24 | 10.75 | 96.91 | 88.29 | 16.49 | 96.98 | 91.98 | 10.60 |
| $1 - \epsilon = 0.9$: | (84.69) | (84.33) | |
| Random                | 94.52 | 89.68 | 19.21 | 93.94 | 85.44 | 21.47 | 94.27 | 89.28 | 19.01 |
| $\mathcal{H}(p_k)$   | 94.49 | 91.31 | 9.91 | 93.67 | 86.41 | 16.29 | 94.54 | 90.80 | 13.08 |
| $p_k^{\text{diff}}$  | 94.49 | 91.31 | 9.91 | 93.67 | 86.53 | 16.11 | 94.02 | 90.56 | 10.69 |
| $p_k^{\text{max}}$ (SM) | 94.49 | 91.31 | 9.91 | 93.68 | 86.44 | 16.13 | 94.05 | 90.76 | 11.01 |
| Meta                  | 94.40 | 90.45 | 8.80 | 93.74 | 86.17 | 15.09 | 94.08 | 89.72 | 8.88 |
Figure C.1: Distribution of exit layers per tolerance level $\epsilon$ (dev sets) with our Shared/ Meta model. See Figure 4 for IMDB.
Figure D.1: Development set results with an Albert-base 12-layers model as $F$. 
Exit layer & Gold label & Input \\
\hline
IMDB (Maas et al., 2011) & & \\
1 & Pos & Without question, film is a powerful medium, more so now than ever before, due to the accessibility of DVD/video, which gives the filmmaker the added assurance that his story or message is going to be seen by possibly millions of people. […] \\
4 & Neg & This movie was obscenely obvious and predictable. The scenes were poorly written and acted even worse. \\
10 & Pos & here in Germany it was only shown on TV one time. today, as everything becomes mainstream, it’s absolute impossible, to watch a film like this again on the screen. maybe it’s the same in USA […] \\
15 & Neg & I tried to be patient and open-minded but found myself in a coma-like state. I wish I would have brought my duck and goose feather pillow… […] \\
20 & Neg & Hypothetical situations abound, one-time director Harry Ralston gives us the ultimate post-apocalyptic glimpse with the world dead, left in the streets, in the stores, and throughout the landscape, sans in the middle of a forgotten desert. […] \\
VitaminC (Schuster et al., 2021) & & \\
3 & Sup & Claim: Another movie titled The SpongeBob Movie: Sponge on the Run is scheduled for release in 2020. Evidence: A second film titled The SpongeBob Movie : Sponge Out of Water was released in 2015, and another titled The SpongeBob Movie: Sponge on the Run is scheduled for release in 2020. \\
5 & Sup & Claim: Julie Bishop offered a defence of her nation’s intelligence cooperation with America. Evidence: The Australian Foreign Minister Julie Bishop stated that the acts of Edward Snowden were treachery and offered a staunch defence of her nation’s intelligence co-operation with America. \\
10 & NEI & Claim: The character Leslie hurts her head on the window in the film 10 Cloverfield Lane. Evidence: Michelle realizes Howard was right and returns his keys. \\
15 & Sup & Claim: Halakha laws are independent of being physically present in the Land of Israel. Evidence: The codification efforts that culminated in the Shulchan Aruch divide the law into four sections, including only laws that do not depend on being physically present in the Land of Israel. \\
20 & Sup & Claim: Germany has recorded less than 74,510 cases of coronavirus , including under 830 deaths. Evidence: 74,508 cases have been reported with 821 deaths and approximately 16,100 recoveries. \\
24 & NEI & Claim: For the 2015-16 school year , the undergraduate fee at USF is under $43,000. Evidence: Undergraduate tuition at USF is $44,040 for the 2016-17 school year. \\
AG News (Gulli, 2004; Zhang et al., 2015) & & \\
1 & Business & Crude Oil Rises on Speculation Cold Weather May Increase Demand Crude oil futures are headed for their biggest weekly gain in 21 months […] \\
5 & Sports & NHL Owner Is Criticized for Talking of Replacement Players The day before the regular season was supposed to open […] \\
10 & World & North Korea Says the Tyrant is Bush, not Kim North Korea says it sees no reason to join a working-level meeting with the United States […] \\
15 & World & Scotch Whisky eyes Asian and Eastern European markets (AFP) AFP - A favourite tipple among connoisseurs the world over, whisky is treated with almost religious reverence on the Hebridean […] \\
20 & Business & Arthritis drug withdrawn after trial A prescription painkiller used by more than 250,000 Australians to treat arthritis has been withdrawn from sale after a clinical trial found it doubled the risk […] \\
24 & Sci/Tech & Airbus drops out of Microsoft appeal Aircraft builder withdraws its request to intervene in Microsoft’s antitrust appeal; Boeing also forgoes intervention. \\
STSB (Cer et al., 2017) & & \\
10 & 0.6 & Sent. 1: A child wearing blue and white shorts is jumping in the surf. Sent. 2: A girl wearing green twists something in her hands. \\
15 & 2.8 & Sent. 1: Saudi Arabia gets a seat at the UN Security Council Sent. 2: Saudi Arabia rejects seat on UN Security Council \\
20 & 4.2 & Sent. 1: a small bird sitting on a branch in winter. Sent. 2: A small bird perched on an icy branch. \\
24 & 3.0 & Sent. 1: It depends entirely on your company and your contract. Sent. 2: It depends on your company. \\
\hline
Table E.1: Number of Transformer layers used for example inputs from the task’s test sets with our Shared/Meta CAT with a tolerance level of $\epsilon = 0.1$