

# ADAPTIVE TEST-TIME INTERVENTION FOR CONCEPT BOTTLENECK MODELS

**Matthew Shen\***

Department of Statistics  
Columbia University  
ms7079@columbia.edu

**Aliyah R. Hsu\***

Department of EECS  
UC Berkeley  
aliyahhsu@berkeley.edu


**Abhineet Agarwal**

Department of Statistics  
UC Berkeley  
aa3797@berkeley.edu

**Bin Yu**

Department of Statistics, EECS  
Center for Computational Biology  
UC Berkeley


## ABSTRACT

Concept bottleneck models (CBM) aim to improve model interpretability by predicting human level “concepts” in a bottleneck within a deep learning model architecture. However, how the predicted concepts are used in predicting the target still either remains black-box or is simplified to maintain interpretability at the cost of prediction performance. We propose to use Fast Interpretable Greedy Sum-Trees (FIGS) to obtain Binary Distillation (BD). This new method, called FIGS-BD, distills a binary-augmented concept-to-target portion of the CBM into an interpretable tree-based model, while maintaining the competitive prediction performance of the CBM teacher. FIGS-BD can be used in downstream tasks to explain and decompose CBM predictions into interpretable binary-concept-interaction attributions and guide adaptive test-time intervention. Across 4 datasets, we demonstrate that our adaptive test-time intervention identifies key concepts that significantly improve performance for realistic human-in-the-loop settings that only allow for limited concept interventions. All code is made available on Github. 

## 1 INTRODUCTION

Deep learning (DL) has achieved impressive performance in various domains such as computer vision (CV) and natural language processing (NLP). Despite their success, DL models are often uninterpretable. Concept bottleneck models (CBMs) (Koh et al., 2020) aim to improve the interpretability of DL models by explaining predictions in terms of human-understandable “concepts”. CBMs can functionally be decomposed into two models: an input-to-concept model and a concept-to-target (CTT) model. Prior CBM work typically uses a linear CTT model for interpretability (Koh et al., 2020; Wong et al., 2021; Ludan et al., 2024). This limits the expressivity of the overall CBM, hurting downstream performance which instead requires CTT models that can capture more complex relationships between concepts. CBMs, especially with practitioner intervention (i.e., check correctness and edit prediction if necessary), have the potential to improve the trustworthiness and usability of models for cases like medical diagnosis (Oikarinen et al., 2023; Yuksekgonul et al., 2023). However, current concept intervention work does not account for difficulties of interventions in high pressure environments with practitioners lacking full domain experience: a surprisingly common scenario where machine learning could be most effectively utilized.

In this work, we address the lack of CTT model interpretability in all concept settings, especially when using complex models to capture complicated CTT relationships (i.e. NLP). We propose the distillation of the CTT portion of the CBM (CTT CBM) with an interpretable Fast Greedy

 <https://github.com/mattyshen/adaptiveTTI>

\*Equal contribution.

Sum-Trees (FIGS) model (Tan et al. 2025). This allows human practitioners to understand the predictions made by the CTT CBM through a sum of contributions depending on interactions of concepts. The FIGS model, through its construction, also adaptively proposes and ranks concepts that are of highest priority for a practitioner to intervene on. The proposed distillation and adaptive test-time intervention process is visualized in Figure 1.

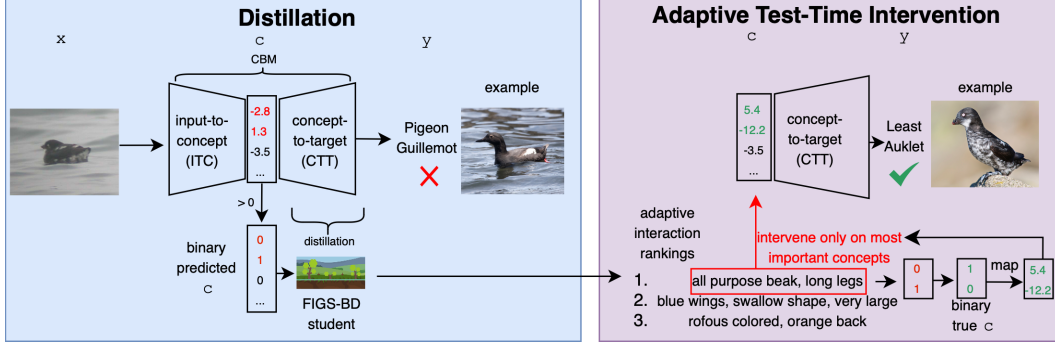


Figure 1: The CBM incorrectly identifies “long legs” in the image, perhaps due to the spurious correlations between water and long legged birds like seagulls. FIGS adaptive test-time intervention (ATTI) recommends a small number (2) of concepts based on a binarization of predicted concepts (including “long legs”) to intervene on, which results in the correct prediction.

## 2 RELATED WORK

### 2.1 CONCEPT MODELS

To improve model interpretability, models can be bottlenecked on human-level “concepts,” popularized by Koh et al. (2020). The usage of concepts to understand models has expanded to analyzing models post-hoc (Yuksekgonul et al. 2023), using other models (i.e. LLMs) or adapting models to iteratively generate and refine concepts for tasks (Oikarinen et al. 2023; Schrodi et al. 2024; Chen et al. 2019; Li et al. 2024; Ludan et al. 2024). Some concept models further learn soft rules (Vemuri et al. 2024) or (decision tree) structures (Nauta et al. 2021), using the predicted concepts to improve interpretability and practitioner usage. Xu et al. (2024) propose energy based CBMs to address limitations of CBMs in capturing nonlinear interactions, and similarly recognize the lack of a principled approach to test-time intervention.

### 2.2 KNOWLEDGE AND MODEL DISTILLATION

In knowledge and model distillation, introduced by Hinton et al. (2015), a compact student model is trained on the predictions of a larger, more complex teacher model to improve inference speed, computation, or even interpretability, while maintaining competitive predictive performance (Jiao et al. 2020). Having an interpretable model that mimics a complex model through distillation can increase the trustworthiness of complex models, streamlining their use into real-life environments.

## 3 FIGS BINARY DISTILLATION – FIGS-BD

We utilize the Fast Interpretable Greedy Sum-Trees (FIGS) algorithm (Tan et al. 2025) to distill the CTT CBMs. We modify the original FIGS algorithm by restricting the maximum depth of the trees learned to maintain interpretability and introduce a multi-output variant to distill the soft-labels (i.e. target logits or probabilities) of CTT CBMs. The FIGS composition of a flexible, yet upper bounded, number of trees and “rules” is inherently interpretable, and practitioners can thus understand predictions made by the (CTT) CBM (and the FIGS student model) as a sum of interactions between concepts. In traditional CBMs, predicted concepts are often logits, which are highly uninterpretable and bring about unnecessary uncertainty to practitioners. An “on” or “off”

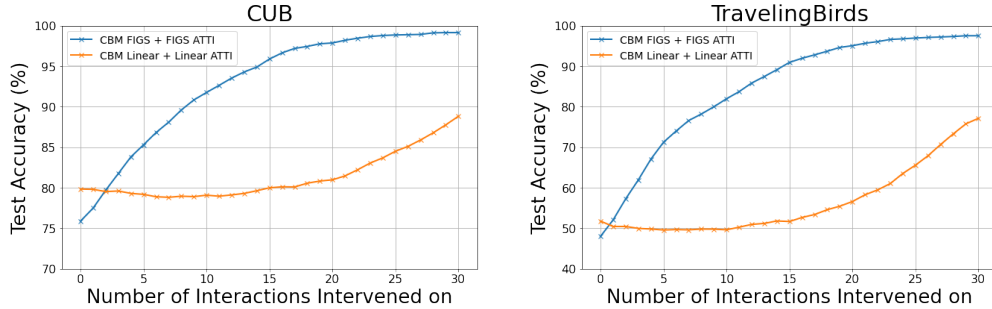


Figure 2: Effectiveness of adaptive test-time interventions for different concept-to-target models. Note the  $x$ -axis enumerates the number of *interactions* (of at most 3 concepts) intervened on.

binary representations of concepts alleviates this uncertainty and lack of interpretability. Thus, we binarize CBM predicted concepts with data-driven (minimize distance between true concepts) or interpretable ( $> 0$ ) thresholds, and distill the CTT CBM using these binary concepts, (teacher) predicted target logits, and FIGS, which we call FIGS-BD.

**Why FIGS?** Predicting targets from binary concepts constitutes learning a Boolean function  $f : \{0, 1\}^d \rightarrow \mathbb{R}$ . All Boolean functions can be expressed as Fourier series (Spiro, 2016). Learning this Fourier series exactly requires exponential samples and time; FIGS-BD instead greedily approximates  $f$  by constructing a sum of shallow trees.

#### 4 DATASETS AND TEACHER MODELS

Our experiments contain two tasks: CV and NLP. For CV, we train CBMs (Koh et al., 2020) on the Caltech-UCSD Birds-200-2011 (CUB) dataset (Wah et al., 2011) and the TravelingBirds (Koh et al., 2020) dataset, which is a variant of CUB where the image backgrounds associated with each bird class are changed from train to test time. CUB and TravelingBirds both pose as challenging prediction tasks with a high number of classes, while TravelingBirds also showcases a distribution shift from train to test time. For NLP, we train LLM-based Text Bottleneck Models (TBMs) (Ludan et al., 2024) on the AGNews topic classification dataset (Zhang et al., 2015) and the CEBaB restaurant reviews (Abraham et al., 2022) dataset (regression task). These two datasets are deemed to have complicated concept interactions in nature that could not be captured in previous TBM work with a linear CTT model (Ludan et al., 2024). More details of experiments are in Appendix A.1

#### 5 DISTILLATION AND PREDICTION PERFORMANCE

Table 1 displays the best test performing CBM/TBM models (with CTT model specified), as well as FIGS and XGBoost (Chen & Guestrin, 2016) student models’ test prediction performance on the CUB, TravelingBirds, AGNews, and CEBaB datasets. A complete table, with other comparative baselines (decision tree and random forest), is in Appendix A.3. Depending on the dataset, the relationship between concept and target can either be very simple or very complex. CUB and TravelingBirds have a fairly linear CTT relationship. For AGNews and CEBaB, complex Transformer models capture the CTT relationship the best, necessitating distillation to improve interpretability and prediction understanding. As evident in the small difference between teacher and student model prediction performance, FIGS-BD is distilling effectively, even in out-of-sample data. FIGS-BD achieves over 92.5 % of the performance of its teacher CBM on the test sets of all surveyed datasets, while generalizing better than the original CBM in some cases (CEBaB). FIGS-BD performs closely with XGBoost in the CUB and CEBaB datasets, and even outperforms XGBoost in the AGNews and TravelingBirds datasets despite being smaller (significantly less rules) and more interpretable (XGBoost fits a separate model for each class, while FIGS-BD fits a single multi-output model).

\*Following the definition in Section 1 a TBM is also a CBM. However we refer to the models used in the NLP tasks as TBMs in the following sections to differentiate from the CBMs used in the CV tasks.

Table 1: Best CBM/TBM test prediction performance with FIGS-BD and XGBoost student models across the 4 datasets. “Teacher Pred” and “Student Pred” denote teacher and student test prediction performance, respectively.

Dataset	Teacher	Student	Teacher Pred	Student Pred
CUB (Acc %)	CBM Linear	FIGS-BD	79.8	75.9
	-	XGBoost	-	75.9
TravelingBirds (Acc %)	CBM Linear	FIGS-BD	51.8	47.9
	-	XGBoost	-	47.7
AGNews (Acc %)	TBM Transformer	FIGS-BD	89.6	88.8
	-	XGBoost	-	88.0
CEBaB (R-squared)	TBM Transformer	FIGS-BD	0.868	0.871
	-	XGBoost	-	0.877

## 6 ADAPTIVE TEST-TIME INTERVENTION USING FIGS-BD

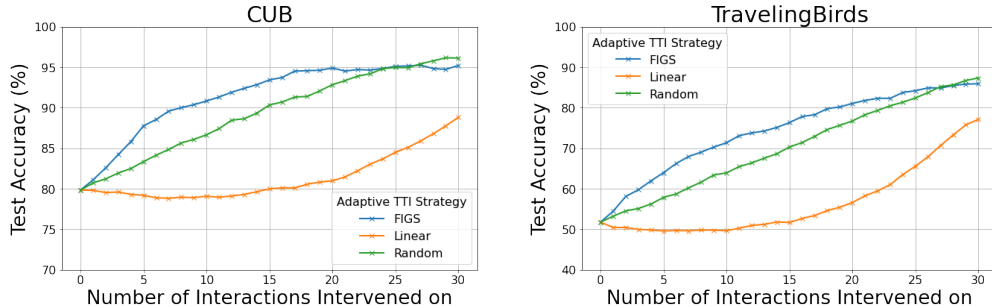


Figure 3: Performance of CBM linear with adaptive test-time interventions for concepts suggested by different CTT models. FIGS ATTI greatly out-performs Linear ATTI.

In high-stakes environments (e.g., emergency rooms), practitioners cannot intervene across all concepts but rather can only do so for a limited number of concepts. In such scenarios, identifying an important ranking of concepts is crucial for accurate prediction. In this subsection, we consider the task: *adaptive test-time intervention (ATTI)* in which a human is allowed to intervene on a small number of concepts for a given test-example. We show how FIGS-BD can be used to adaptively rank the most important concepts for a human to validate before prediction.

We propose constructing a sample-specific ranking of concepts based on the highest variance of absolute predictions (across the target dimension) path, from where the concepts are identified, that the sample falls down. Algorithm 1 describes this process in pseudo code. Similarly, for linear CTT portions, we propose ranking concepts based on the highest variation of absolute values of the product of fitted coefficients and predicted concept values. We believe that higher variance (across the target dimension) represents “volatile” contributions that are the most important to intervene on. More details can be found in Appendix A.2

**Quantitative prediction improvement on CUB and TravelingBirds** We conduct an experiment where a practitioner is allowed to intervene on the top- $k$  interactions of concepts for a test sample recommended by various TTI methods. We consider top concepts recommended by FIGS-BD, a linear CTT, as well as random selection. We plot the results in Figure 3. FIGS identifies concepts that are much more relevant for making a correct prediction, indicating its utility in identifying relevant concepts for humans to validate.

Additionally, we conduct an ablation study comparing the original linear CTT model (CBM Linear) with linear ATTIs and the FIGS-BD CTT model (CBM FIGS) with FIGS ATTIs. We plot the results in Figure 2. The FIGS-BD CTT model quickly surpasses the linear with a practitioner’s interventions, reaching drastically higher test accuracy %s with a moderate to large number of interventions. Specifically, in as few as 3 and 1 interaction interventions for CUB and TravelingBirds,

respectively, the FIGS-BD CTTs outperforms linear CTTs. This highlights the impact of editing with binary values (rather than with predicted training data quantiles) and the effectiveness of FIGS ATTI. On TravelingBirds, FIGS-BD requires far less interventions (7) to reach the same accuracy as the maximum intervened on (30) linear model, potentially further disentangling the detrimental spurious correlation that was propagated into the original linear CTT model.

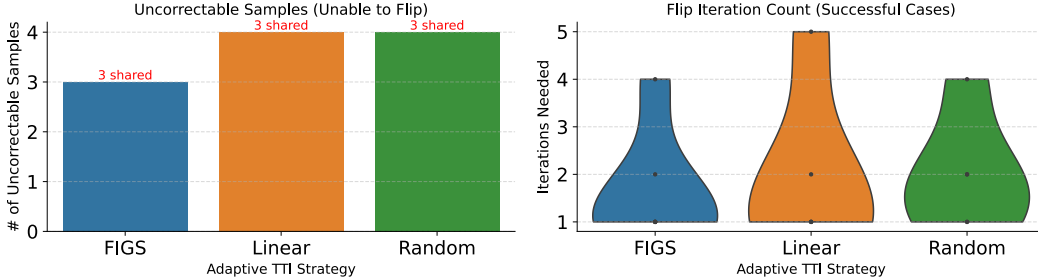


Figure 4: **Left:** number of uncorrectable samples of each intervention method. **Right:** count of iterations of intervention needed of each method to flip a wrong prediction into a correct one.

**Effectiveness for correcting model prediction on AGNews and CEBaB** Unlike CUB and TravelingBirds, AGNews and CEBaB lack human-labeled concepts. To address this, we manually annotate a small set of misclassified samples from these datasets with concept labels<sup>\*</sup>. We then evaluate different intervention methods by measuring how many iterations of interventions it takes to “flip” an incorrect prediction to a correct one, intervening sequentially via ATTI interactions rankings.

Figure 4 shows results on AGNews and CEBaB (combined) using the linear CTT model with various ATTI strategies. FIGS ATTI consistently achieves successful flips with fewer iterations compared to other ATTIs. Moreover, it results in fewer uncorrectable samples (i.e., samples for which all recommended interventions fail to correct the prediction), and its uncorrectable samples are a strict subset of those from other methods, indicating that it successfully recovers some cases others cannot. In contrast, linear ATTI even underperforms random ATTI, suggesting that linear models struggle to recommend reliable concepts for intervention.

As a case study, we highlight an example from CEBaB where FIGS ATTI was the *only* method able to correct the model’s incorrect prediction, requiring just *one* intervention. The linear CTT model initially misclassified the review "My dining experience was one of the best. The food and service was outstanding. Everyone was very friendly just could have turned down the volume of the music a little." with a rating of 5 instead of the ground-truth rating of 4. The model overemphasized the concept "Customer Expectations", which was not present in the review. FIGS ATTI correctly identified "Customer Expectations", "Overall Satisfaction", and "Service Quality" as the most critical concepts for intervention, enabling a human to downweight the erroneous concept and produce the correct rating. In contrast, the other methods failed to surface this issue in their recommended interactions. In short, FIGS-BD ATTI’s superior performance results from its ability to identify concept interactions are crucial in determining between two competing classes effectively for human intervention.

## 7 CONCLUSION

In this paper, we propose FIGS-BD: an algorithm to distill binary-augmented concept-to-target portions of CBMs to interpret their predictions as contributions of concept interactions. From the FIGS-BD student model, we introduce adaptive test-time introduction, which requires CBMs to propose a small number of concepts to be validated before prediction. FIGS-BD identifies more relevant concepts for accurate prediction. Future work involves extension to post-hoc CBMs, further empirical evaluation, and counterfactual predictions with FIGS.

<sup>\*</sup> Annotations were performed by three PhD students specializing in statistics or computer science.

## ACKNOWLEDGEMENTS

We gratefully acknowledge partial support from NSF grants DMS-2209975 and DMS-2413265, NSF grant 2023505 on Collaborative Research: Foundations of Data Science Institute (FODSI), the NSF and the Simons Foundation for the Collaboration on the Theoretical Foundations of Deep Learning through awards DMS-2031883 and 814639, NSF grant MC2378 to the Institute for Artificial CyberThreat Intelligence and Operation (ACTION), NIH grant R01GM152718 (DMS/NIGMS), a Berkeley Deep Drive (BDD) Grant from BAIR and a Dean’s fund from CoE, at UC Berkeley.

## REFERENCES

- Eldar David Abraham, Karel D’Oosterlinck, Amir Feder, Yair Ori Gat, Atticus Geiger, Christopher Potts, Roi Reichart, and Zhengxuan Wu. Cebab: Estimating the causal effects of real-world concepts on nlp model behavior, 2022. URL <https://arxiv.org/abs/2205.14140>
- Leo Breiman. Random forests. 45(1):5–32, 2001. doi: 10.1023/A:1010933404324.
- Leo Breiman, Jerome Friedman, RA Olshen, and Charles J Stone. *Classification and regression trees*. Chapman and Hall/CRC, 1984.
- Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. This looks like that: Deep learning for interpretable image recognition, 2019. URL <https://arxiv.org/abs/1806.10574>
- Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, pp. 785–794. ACM, August 2016. doi: 10.1145/2939672.2939785. URL <http://dx.doi.org/10.1145/2939672.2939785>
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL <https://arxiv.org/abs/1503.02531>
- Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding, 2020. URL <https://arxiv.org/abs/1909.10351>
- Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models, 2020. URL <https://arxiv.org/abs/2007.04612>
- Aaron J. Li, Robin Netzorg, Zhihan Cheng, Zhuoqin Zhang, and Bin Yu. Improving prototypical visual explanations with reward reweighing, reselection, and retraining, 2024. URL <https://arxiv.org/abs/2307.03887>
- Josh Magnus Ludan, Qing Lyu, Yue Yang, Liam Dugan, Mark Yatskar, and Chris Callison-Burch. Interpretable-by-design text understanding with iteratively generated concept bottleneck, 2024. URL <https://arxiv.org/abs/2310.19660>
- Meike Nauta, Ron van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition, 2021. URL <https://arxiv.org/abs/2012.02046>
- Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models, 2023. URL <https://arxiv.org/abs/2304.06129>
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis,



- Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rameesh Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL <https://arxiv.org/abs/2303.08774>
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85): 2825–2830, 2011. URL <http://jmlr.org/papers/v12/pedregosa11a.html>
- Simon Schrodli, Julian Schur, Max Argus, and Thomas Brox. Concept bottleneck models without predefined concepts. *arXiv preprint arXiv:2407.03921*, 2024.
- Chandan Singh, Keyan Nasseri, Yan Shuo Tan, Tiffany Tang, and Bin Yu. imodels: a python package for fitting interpretable models. *Journal of Open Source Software*, 6(61):3192, 2021. doi: 10.21105/joss.03192. URL <https://doi.org/10.21105/joss.03192>
- Sam Spiro. Fourier analysis of boolean functions, 2016. URL <https://math.uchicago.edu/~may/REU2016/REUPapers/Spiro.pdf>
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Re-thinking the inception architecture for computer vision, 2015. URL <https://arxiv.org/abs/1512.00567>

- Yan Shuo Tan, Chandan Singh, Keyan Nasser, Abhinav Agarwal, James Duncan, Omer Ronen, Matthew Epland, Aaron Kornblith, and Bin Yu. Fast interpretable greedy-tree sums. *Proceedings of the National Academy of Sciences*, 122(7):e2310151122, 2025. doi: 10.1073/pnas.2310151122. URL <https://www.pnas.org/doi/abs/10.1073/pnas.2310151122>.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL <https://arxiv.org/abs/1706.03762>.
- Deepika Vemuri, Gautham Bellamkonda, and Vineeth N Bellamkonda. Enhancing concept-based learning with logic, 2024. URL <https://openreview.net/pdf?id=FUZYp83y4M>.
- Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset, August 2011.
- Eric Wong, Shibani Santurkar, and Aleksander Madry. Leveraging sparse linear layers for debuggable deep networks, 2021. URL <https://arxiv.org/abs/2105.04857>.
- Xinyue Xu, Yi Qin, Lu Mi, Hao Wang, and Xiaomeng Li. Energy-based concept bottleneck models: Unifying prediction, concept intervention, and probabilistic interpretations, 2024. URL <https://arxiv.org/abs/2401.14142>.
- Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models, 2023. URL <https://arxiv.org/abs/2205.15480>.
- Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NIPS*, 2015.

## A APPENDIX

### A.1 MODEL ARCHITECTURES AND HYPERPARAMETERS

For all datasets and CBM/TBM models, we utilize code and the majority of architectures provided by the authors of respective papers. For more complex concept-to-target (CTT) portions of the CBM, we modified provided code and scripts to train and evaluate the model. For FIGS models, we have contributed to the `imodels` (Singh et al., 2021) package (specifically, the FIGS implementation) to restrict the maximum depth of trees, handle multi-output prediction tasks, and create cross validation (CV) models. We have then adapted that model for ATTIs.

#### A.1.1 CUB AND TRAVELING BIRDS

The Caltech-UCSD Birds-200-2011 (CUB) dataset (Wah et al., 2011) and the TravelingBirds (Koh et al., 2020) dataset contain  $n = 11,788$  photos of birds with 200 bird class labels. Every observation in the dataset comes with human-labelled annotations regarding concepts present in the image, which facilitates our ATTI experiments. We reduce the number of concepts used in the same procedure as described in Koh et al. (2020). We utilize the code, instructions, and some trained models provided by Koh et al. (2020). We modify parts of their Github repository to incorporate more complex concept-to-target models. Specifically, we include MLP with 1 hidden layer, MLP with 2 hidden layers, and a simple Transformer model (encoder-only). All MLPs have the same hidden size, set to be 250 for CUB and TravelingBirds. The Transformer model utilizes multi-headed attention (Vaswani et al., 2023) with 4 heads, a MLP with 1 hidden layer of hidden size 250, and then a linear classifier layer. For the input-to-concept portion of the CBM, we utilize the Inception V3 (Szegedy et al., 2015) model, and for the overall model, utilize the overall Joint training process with  $\lambda = 0.01$ . All hyperparameters regarding training are the same as in Koh et al. (2020). Due to the complicated 200 class prediction task posed by CUB and TravelingBirds, we utilize a FIGS CV model to determine the hyperparameters that result in the strongest FIGS-BD model. We use an interpretable rule of  $> 0$  (anpositive concept prediction results in 1, negative results in 0) to binarize concept features before FIGS distillation. We search over [125, 200] rules, [30, 40] trees, and [3, 4] max depth. For CV results in Table 1 the post cross-validation fitted FIGS-BD model results in 200 rules, 30 trees, and max depth of 3 for both CUB and TravelingBirds.



### A.1.2 AGNews AND CEBaB

AGNews contains  $n = 7,600$  news articles and 4 class labels (world, sports, business, and sci/tech) for news topic classification. CEBaB contains  $n = 1,713$  restaurant views and their corresponding ratings (1-5) from customers as labels, and we formulate it as a regression task. For both datasets, we randomly split them into  $n = 1,500$  train set and  $n = 250$  test set for training and evaluation. To be comparable to the original TBM (Ludan et al., 2024) paper, we use GPT-4 (GPT-4-0613) (OpenAI et al., 2024) as the underlying LLM for concept generation and concept measurement in the input-to-concept portion of the TBMs. The original TBM code uses Scikit-learn (Pedregosa et al., 2011) for training linear regression (regression task) and logistic regression (classification task) for the concept-to-target portions of the TBMs. We modify parts of their code to incorporate more complex concept-to-target models. Specifically we include MLP with 1 hidden layer, MLP with 2 hidden layers, and a simple Transformer model (encoder-only). All MLPs have the same size set to be 50 for both AGNews and CEBaB datasets. The Transformer model utilizes two blocks of multi-headed attention (4 heads) + MLP with 1 hidden layer (hidden size 52) module, and then a linear classifier layer. All hyperparameters regarding training are the same as in (Ludan et al., 2024), except for the refinement trial size, which we set to be 500 for training the more complicated CTT models (MLPs and the simple transformer). We use one-hot-encoding to binarize concept features before FIGS distillation. We search over [100, 200, 250] rules, [20, 30, 50] trees, and [3, 4] max depth. For the NLP results in Table 1 the post cross-validation fitted FIGS-BD model results in 154 rules, 50 trees, and max depth of 3 for both CEBaB and AGNews.

### A.2 ADAPTIVE TTI INTERVENTIONS

---

**Algorithm 1** FIGS-BD ATTI algorithm

---

```

1: FIGSBD.ATTI( $f_{\text{FIGS}}$ : FIGS-BD model,  $x$ :  $\mathbb{R}^{n_{\text{concepts}}}$ )
2: all_trees = trees( $f_{\text{FIGS}}$ )
3: tree_predictions = []
4: tree_paths = []
5: for tree in all_trees do
6:   tree_prediction.append(tree.predict( $x$ ))
7:   tree_paths.append(pathtree( $x$ ))
8: end for
9: predictions_and_paths = zip(tree_predictions, tree_paths)
10: rankings = sort(predictions_and_paths, lambda  $x_{\text{pred}}$  : var( $|x_{\text{pred}}|$ ) or max( $|x_{\text{pred}}|$ ))
11: return rankings

```

---

FIGS-BD ranks interactions of concepts that are embedded in the structure of its collection of trees. The ranking procedure is described in pseudo code in Algorithm 1. Thus, every set or interaction of concepts to be intervened on in Figure 3 are of size maximum depth of grown tree. Note that this is not always equal to the maximum depth hyperparameter of the model, as the FIGS model does not have to grow to full depth. Additionally, concepts are re-used in some learned interactions, so intervention is not as effective after many interventions have occurred (and are thus the most impactful for the earlier sets of interactions intervened on). For each observation, these interactions of varying size are ranked based on a heuristic function (variance of absolute value of multi-output prediction and maximum of absolute value for 1-dimensional output prediction). For random ATTI, we randomly choose concepts without replacement and group/parse them of corresponding size to every FIGS ATTI to make them comparable to FIGS ATTI. For linear ATTI, we rank the  $n_{\text{concepts}}$  concepts based on variance of absolute value of product of concept prediction and concept coefficient, and group/parse them of corresponding size to every FIGS ATTI to make them comparable to FIGS ATTI. Note that when talking about variance, we refer to the variance of predictions across the multi-output target dimension.

For all of our experiments, the best FIGS-BD student models grow to depths of 3, which means that the maximum size of every interaction or cluster of concepts intervened on is 3.

For CUB and TravelingBirds, as done by (Koh et al., 2020) for interventions, we replace the predicted concept values with the 5th quantile and 95th quantile of the predicted concept in the

training data if the true concept is 0 and 1 for the original CBM (linear CTT), respectively. This is denoted as “map” in Figure 11. This can result in prediction performance degrading as replacing predicted values for a specific instance with training values, even if the pre-intervention and post-intervention concept values agree in some way (one could perhaps argue for equivalence in sign meaning an agreement, but there is no exact way without uncertainty to determine if a CBM predicted a concept correctly).

### A.3 FULL CBM AND STUDENT MODEL PREDICTION AND DISTILLATION RESULTS

Table 2 contains all teacher and student test prediction performances across a variety of teacher models and selection of student (regression) models: FIGS, XGBoost (Chen & Guestrin 2016), Random Forest (RF) (Breiman 2001), and Decision Tree (DT) (Breiman et al. 1984). The teacher models vary in their concept-to-target portion, in which we consider Linear, MLP1, MLP2, and Transformer concept-to-target models. Note that FIGS-BD was trained using cross-validation, meaning that it is likely that if the teacher model is more complex, the FIGS-BD student model consists of more trees, more rules, and more depth. For CUB and TravelingBirds, we restricted XGBoost and RF to 30 trees (the same amount as the cross-validation-chosen FIGS-BD model). We depth-restricted XGBoost, RF, and DT to 3 (same as cross-validation chosen FIGS-BD model), 7 or 8, and 7 or 8, respectively, and chose the best performing model. We choose depth 7 or 8 because there are 200 classes in the CUB and TravelingBird tasks, so we need enough expressivity (and leaf nodes:  $2^7 = 128$ ,  $2^8 = 256$ ) to achieve strong performance. For AGNews and CEBaB, we restricted XGBoost and RF to 50 trees, and depth-restricted XGBoost, RF, and DT to 3 (same as cross-validation chosen FIGS-BD model), 2 or 3, and 2 or 3, respectively, following the same logic. The results displayed consist of XGBoost, RF, and DT of depth 3, 8, and 8, respectively for CUB and TravelingBirds, and of depth 3 for all three models for AGNews and CEBaB.

On all datasets, XGBoost displays strong performance, but we note that XGBoost was not restricted in terms of number of rules (only restricted in depth and tree) and XGBoost also grows a separate estimator per class/task, for example, resulting in  $30 \cdot 200 = 6000$  total trees (for CUB and TravelingBirds) with max depth 3. Thus, XGBoost is highly uninterpretable and grows highly inefficient and dense trees. On the other hand, FIGS-BD grows sparser and is a number-of-rules restricted model, consisting of only 30 trees of max depth 3 for CUB and TravelingBirds. RF and DT perform significantly worse than XGBoost and FIGS-BD, while RF is also highly uninterpretable.

Table 2: Full CBM (teacher model) and student model test prediction performance across the datasets. “Teacher Pred” and “Student Pred” denote teacher and student test prediction performance, respectively. Top prediction performance for each dataset and model role (teacher or student) in bold. The second-best student performance is underlined. RF and DT denote Random Forest and Decision Tree, respectively.

Dataset	Teacher	Student	Teacher Pred	Student Pred
CUB (Acc %)	CBM Linear	FIGS-BD	<b>79.8</b>	<b>75.9</b>
	-	XGBoost	-	<b>75.9</b>
	-	RF	-	64.4
	-	DT	-	50.2
	CBM MLP1	FIGS-BD	79.0	<u>73.7</u>
	-	XGBoost	-	<b>74.1</b>
	-	RF	-	65.3
	-	DT	-	51.8
	CBM MLP2	FIGS-BD	78.0	<u>72.7</u>
	-	XGBoost	-	<b>74.2</b>
	-	RF	-	65.4
	-	DT	-	48.0
	CBM Transformer	FIGS-BD	77.4	<u>72.2</u>
	-	XGBoost	-	<b>73.0</b>
TravelingBirds (Acc %)	-	RF	-	66.2
	-	DT	-	51.5
	CBM Linear	FIGS-BD	<b>51.8</b>	<b>47.9</b>
	-	XGBoost	-	<u>47.7</u>
	-	RF	-	38.4
	-	DT	-	28.5
	CBM MLP1	FIGS-BD	49.2	<u>48.5</u>
	-	XGBoost	-	<b>50.1</b>
	-	RF	-	41.5
	-	DT	-	31.5
	CBM MLP2	FIGS-BD	49.6	<u>49.1</u>
	-	XGBoost	-	<b>49.7</b>
	-	RF	-	42.0
	-	DT	-	33.7
AGNews (Acc %)	CBM Transformer	FIGS-BD	47.5	<u>47.1</u>
	-	XGBoost	-	<b>47.2</b>
	-	RF	-	43.4
	-	DT	-	32.2
	TBM Linear	FIGS-BD	84.8	<u>83.2</u>
	-	XGBoost	-	<b>86.8</b>
	-	RF	-	82.8
	-	DT	-	81.2
	TBM MLP1	FIGS-BD	84.4	<u>80.8</u>
	-	XGBoost	-	<b>83.6</b>
	-	RF	-	76.4
	-	DT	-	78.8
	TBM MLP2	FIGS-BD	80.8	<u>79.2</u>
	-	XGBoost	-	<b>82.0</b>
CEBaB (R-squared)	-	RF	-	76.8
	-	DT	-	76.8
	TBM Transformer	FIGS-BD	<b>89.6</b>	<b>88.8</b>
	-	XGBoost	-	<u>88.0</u>
	-	RF	-	83.2
	-	DT	-	83.2
	TBM Linear	FIGS-BD	0.761	<u>0.797</u>
	-	XGBoost	-	<b>0.804</b>
	-	RF	-	0.784
	-	DT	-	0.785
	TBM MLP1	FIGS-BD	0.837	<u>0.873</u>
	-	XGBoost	-	<b>0.882</b>
	-	RF	-	0.864
	-	DT	-	0.863
CEBaB (R-squared)	TBM MLP2	FIGS-BD	0.808	<b>0.833</b>
	-	XGBoost	-	<b>0.833</b>
	-	RF	-	0.813
	-	DT	-	0.812
	TBM Transformer	FIGS-BD	<b>0.868</b>	<u>0.871</u>
	-	XGBoost	-	<b>0.877</b>
	-	RF	-	0.847
	-	DT	-	0.786