PLEX: Towards Reliability using Pretrained Large Model Extensions

Dustin Tran [1] **Jeremiah Liu** [1] **Michael W. Dusenberry** [1] **Du Phan** [1] **Mark Collier** [1] **Jie Ren** [1] **Kehang Han** [1]
Zi Wang [1] **Zelda Mariet** [1] **Huiyi Hu** [1] **Neil Band** [2] **Tim G. J. Rudner** [2] **Karan Singhal** [1] **Zachary Nado** [1]
Joost van Amersfoort [2] **Andreas Kirsch** [2] **Rodolphe Jenatton** [1] **Nithum Thain** [1] **Honglin Yuan** [1 *]
Kelly Buchanan [1 *] **Kevin Murphy** [1] **D. Sculley** [1] **Yarin Gal** [2] **Zoubin Ghahramani** [1] **Jasper Snoek** [1]
Balaji Lakshminarayanan [1]

Abstract

A recent trend in artificial intelligence is the use of pretrained models for language and vision tasks, which have achieved extraordinary performance but also puzzling failures. Probing these models' abilities in diverse ways is therefore critical to the field. In this paper, we explore the *reliability* of models, where we define a reliable model as one that not only achieves strong predictive performance but also performs well consistently over many decision-making tasks involving uncertainty (e.g., selective prediction, open set recognition), robust generalization (e.g., accuracy and proper scoring rules such as log-likelihood on in- and out-of-distribution datasets), and adaptation (e.g., active learning, few-shot uncertainty). We devise 10 types of tasks over 38 datasets in order to evaluate different aspects of reliability on both vision and language domains. To improve reliability, we developed ViT-Plex and T5-Plex, *p*retrained *l*arge model *ex*tensions (PLEX) for vision and language modalities, respectively. Plex greatly improves the state-of-the-art across reliability tasks, and simplifies the traditional protocol as it does not require designing scores or tuning the model for each individual task. We demonstrate scaling effects over model sizes up to 1B parameters and pretraining dataset sizes up to 4B examples. We also demonstrate Plex's capabilities on challenging tasks including zero-shot open set recognition, active learning, and uncertainty in conversational language understanding.[1]

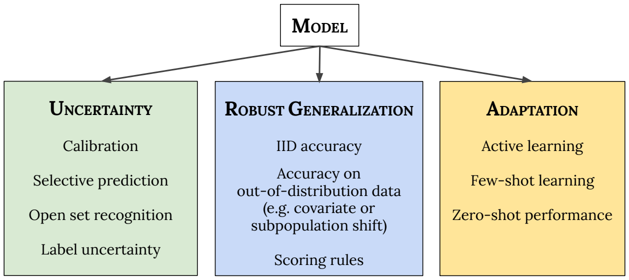


Figure 1. Desiderata for a Reliable model. We propose to simultaneously stress-test the "out-of-the-box" model performance (i.e. the predictive probability distribution $p(y|x)$) across a suite of uncertainty, robust generalization, and adaptation benchmarks, without any customization for individual tasks.

1 Reliability as a Goal for AI

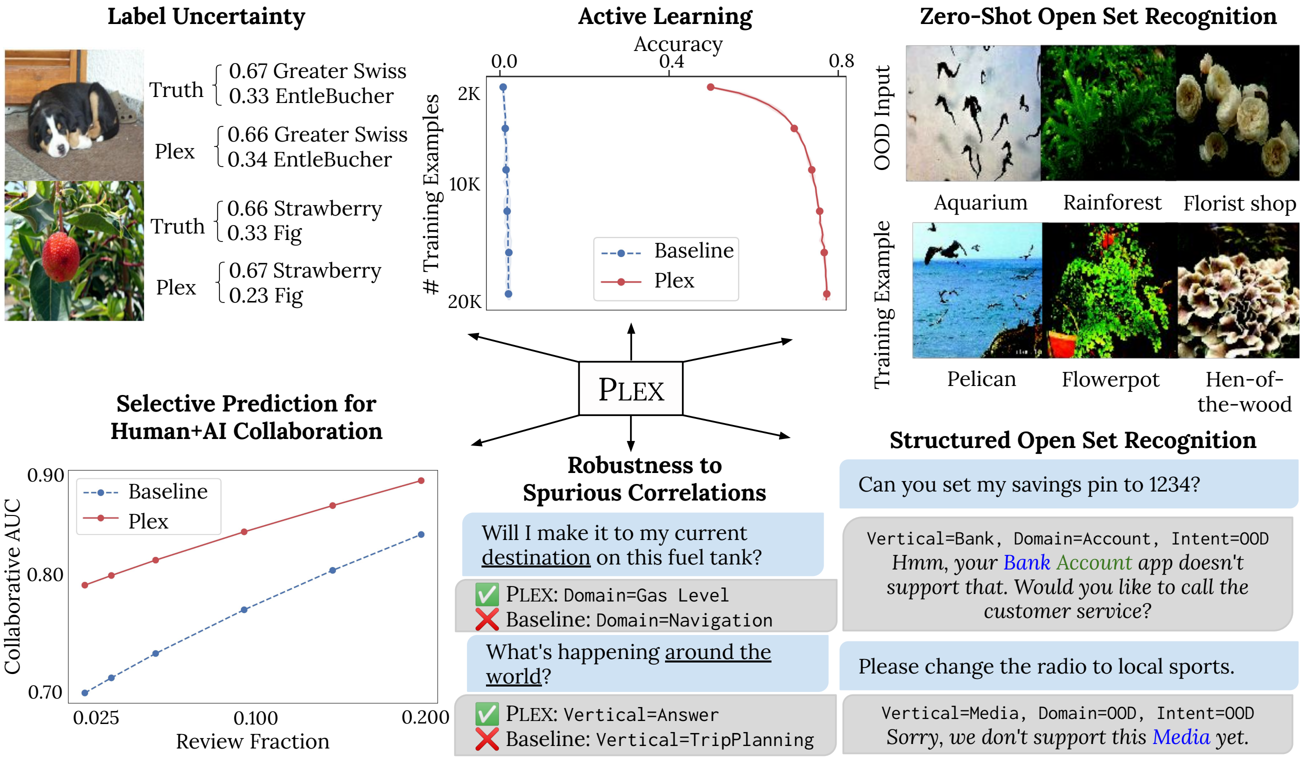
Over the past few years, the deep learning approach to artificial intelligence (AI) has made significant progress on benchmark tasks across domains such as computer vision (Dosovitskiy et al., 2020) and natural language processing (Raffel et al., 2020; Brown et al., 2020). With this progress, there is unfettered excitement about the potential of AI to have a transformative impact. While hypothesizing about this potential is important, we highlight that the tasks where deep learning has been most successful have been carefully devised to fit within narrow boundaries—for example, a focus on predictive performance with test inputs close to the data on which the model was trained.

To go beyond these limitations, we argue that the ability of models to make *reliable* decisions is critical to the deeper integration of AI in the real world. Here, we define reliability as the ability for a model to work consistently across real-world settings. We borrow the term from reliability engineering (Barlow & Proschan, 1975; O'Connor & Kleyner, 2012), a discipline of engineering involving risk assessment, testability, and fault tolerance. Related nomenclature include robustness (Russell et al., 2015), safety (Amodei et al., 2016; Everitt et al., 2018; Hendrycks et al., 2021b), calibration (Dawid, 1982), credibility (D'Amour et al., 2020) and trustworthiness (Avin et al., 2021), each with their own broad and intersecting scopes.

---

[1]Google [2]University of Oxford [*]Work done at Google. Correspondence to: Dustin Tran <trandustin@google.com>.
Presented at ICML Workshop on Principles of Distribution Shift 2022, Copyright 2022 by the author(s).

[1] Code for training & evaluation is open-sourced in Uncertainty Baselines (Nado et al., 2021). Layer and method implementations use Edward2 (Tran et al., 2018).



Will I make it to my current <u>destination</u> on this fuel tank?

✅ PLEX: `Domain=Gas Level`
❌ Baseline: `Domain=Navigation`

What's happening <u>around the world</u>?

✅ PLEX: `Vertical=Answer`
❌ Baseline: `Vertical=TripPlanning`

Can you set my savings pin to 1234?

`Vertical=Bank, Domain=Account, Intent=OOD`
Hmm, your Bank Account app doesn't support that. Would you like to call the customer service?

Please change the radio to local sports.

`Vertical=Media, Domain=OOD, Intent=OOD`
Sorry, we don't support this Media yet.

Figure 2. **Top row: Examples of Plex's capabilities in vision**: **(left)** Label uncertainty in ImageNet ReaL-H, demonstrating the ability to capture the inherent ambiguity of image labels. **(middle)** Active learning on ImageNet1K, displaying Plex's label efficiency compared to a baseline. **(right)** Zero-shot open set recognition on ImageNet1K vs Places365, showing that Plex can distinguish visually similar images without finetuning. **Bottom row: Examples of Plex's capabilities in language**: **(left)** Plex enables human+AI collaboration by improving selective prediction, where the model is given the option to defer a fraction of the test examples to humans. Plex is able to better identify cases where it is likely to be wrong than the baseline. **(middle)** Plex is robust while a baseline latches onto spurious features such as "destination" and "around the world". **(right)** Plex enables structured open set recognition. This provides nuanced clarifications, where Plex can distinguish cases where the request's domain and vertical are supported but the intent is not.

Desiderata for Reliability The majority of machine learning research focuses on measures of performance based on the accuracy on a test set drawn from the same distribution as the training set, the so-called independent and identically distributed (i.i.d.) assumption. However, this does not capture the real-world deployment of AI systems, where often the testing environment is very different from the training environment. The emphasis in our paper is on how reliable an AI system is in such novel scenarios. We posit three general categories of desiderata for reliable AI systems: they should represent their own uncertainty, they should generalize robustly to new scenarios, and their learning procedures should be able to adapt to new data.

Importantly, the aim for a reliable model is to do well in *all* of these areas simultaneously out-of-the-box without requiring any customization for individual tasks (Figure 1):

1. *Uncertainty* involves imperfect or unknown information where it is impossible to exactly describe an existing state (Ghahramani, 2015). Predictive uncertainty quantification allows one to compute optimal decisions (Parmigiani & Inoue, 2009), and enables practitioners to know when to trust the model's predictions, thereby enabling graceful failures when the model is likely to be wrong. In the latter case, which is often referred to as *selective prediction*, the model may defer its prediction to human experts when it is not confident.

2. *Robust Generalization* involves an estimate or forecast about an unseen event (Abraham & Ledolter, 1983; Dawid, 1982). The quality of prediction is typically measured using accuracy (e.g. top-1 error for classification problems and mean squared error for regression problems) and proper scoring rules such as log likelihood and Brier score (Gneiting & Raftery, 2007). In the real world, we care not only about metrics on new data obtained from the same distribution the model was trained on (i.i.d.), but also about *robustness*, as measured by metrics on data under out-of-distribution shifts such as covariate or subpopulation shift.

3. *Adaptation* involves probing the model's abilities over the course of its learning process. Benchmarks typically evaluate on static datasets with pre-defined train-test splits. However, in many applications, we are interested in models that can quickly adapt to new datasets and efficiently learn with as few labeled examples as possible. Examples include few-shot learning (Ravi & Larochelle, 2017), where the model learns from a small set of examples; active learning (Settles, 2009), where the model not only learns but also participates in acquiring the data to learn from; and lifelong learning (Thrun, 1998), where the model learns over a sequence of tasks and must not forget about relevant information for previous tasks.

Contributions First, we define and evaluate reliability in a comprehensive fashion. We use 10 types of tasks in order to capture the three reliability areas—uncertainty, robust generalization, and adaptation—and so that the tasks measure a diverse set of desirable properties in each area. Together the tasks comprise 36 downstream datasets across vision and natural language modalities: 14 datasets for finetuning (including few-shot and active learning-based adaptation) and 22 datasets for out-of-distribution evaluation (Appendix A).

To improve reliability, we develop ViT-Plex and T5-Plex, building on large pretrained models on vision (ViT (Dosovitskiy et al., 2020)) and language (T5 (Raffel et al., 2020)) respectively. We train variants of Plex over multiple model sizes and pretraining dataset sizes on up to 4 billion examples. Figure 3 illustrates Plex's performance on a select set of tasks comparing to existing state-of-the-art, which typically use models specialized for that task. Plex greatly improves the state-of-the-art over the total of 36 datasets. Importantly, Plex achieves impressive performance across all tasks using out-of-the box model output without requiring any custom designing or tuning for each individual task.

2 Tasks for Benchmarking Reliability

We evaluate a model's reliability using 10 types of tasks, which we define below. We selected a broad suite of 36 downstream datasets under the tasks, each ranging from several hundred to a million examples; see Appendix A.

Uncertainty: Selective prediction jointly assesses the predictive performance and quality of uncertainty estimates of a model, by abstaining from making predictions on examples for which a model's predictive uncertainty estimates are above a given threshold and recording predictive accuracy on the remaining examples. We compute two metrics, Calibration AUC and Oracle Collaborative Accuracy (Kivlichan et al., 2021), on 4 image and 10 text datasets. **Open set recognition** assesses how well a model can detect examples belonging to none-of-the training classes. We use AUROC and experiment with maximum softmax probability as the detection score. (We use Mahalanobis distance for zero-shot open set recognition.) **Data uncertainty** measures the uncertainty inherent in the data. An important subset is *label uncertainty*, e.g. the human raters may not agree about the label for ambiguous examples. If this disagreement is encoded as a label distribution, we can directly compare our model's predictive distribution to it. We use two datasets: CIFAR-10H (Peterson et al., 2019) and ImageNet ReaL (Beyer et al., 2020). **Calibration** assesses how well a model's predicted confidence is reflected over a population (Dawid, 1982). We compute expected calibration error (Naeini et al., 2015) on 14 image and 10 text datasets.

Robust Generalization: We assess **in-distribution generalization**, i.e. how well a model can make predictions after finetuning, by examining accuracy, negative log-likelihood, and Brier score on the in-distribution test splits of 5 image and 3 text datasets. With **out-of-distribution data**, we assess how robustly a model's predictions generalize to input distributions it was not trained on. We use the same metrics measured for in-distribution, and we investigate 4 types of out-of-distribution data: covariate shift, semantic (class) shift, data uncertainty, and subpopulation shift.

Adaptation: Few-shot learning assesses how well a model can make predictions downstream with only a few training examples. We use 9 datasets and apply multiple few-shot settings: 1-shot, 5-shot, 10-shot, and 25-shot (x-shot means x examples per class). We also evaluate **few-shot uncertainty**, where we examine calibration, selective prediction, and open set recognition in the few-shot regime. We use all 9 datasets for few-shot learning in order to evaluate calibration and selective prediction, and we use those with OOD datasets (ImageNet and CIFAR-100) for open set recognition. We also perform **zero-shot open set recognition** by using the Mahalanobis distance scoring to detect whether an input is out-of-distribution based on the model's representation layer. **Active learning** assesses how well a model knows what it does not know by selecting informative samples to label using uncertainty. We assess accuracy over a total number of acquired examples and apply *margin sampling* (Settles, 2009) for multi-class uncertainty sampling.

3 PLEX: *P*retrained *L*arge model *Ex*tensions

Plex is the result of an extensive study of the reliability of large pretrained models and their complementarity with existing reliability methods. In particular, ViT Plex and T5 Plex use several key ingredients:

- **Base Transformer architecture.** We adopt the Transformer standard of an alternating sequence of attention and feedforward layers. We build on T5 1.1 (Raffel et al., 2020) for text as a Transformer in an encoder-decoder setup where the raw text is tokenized with SentencePiece, and on Vision Transformer (Dosovitskiy et al., 2020) for images in an encoder-only setup where the raw images

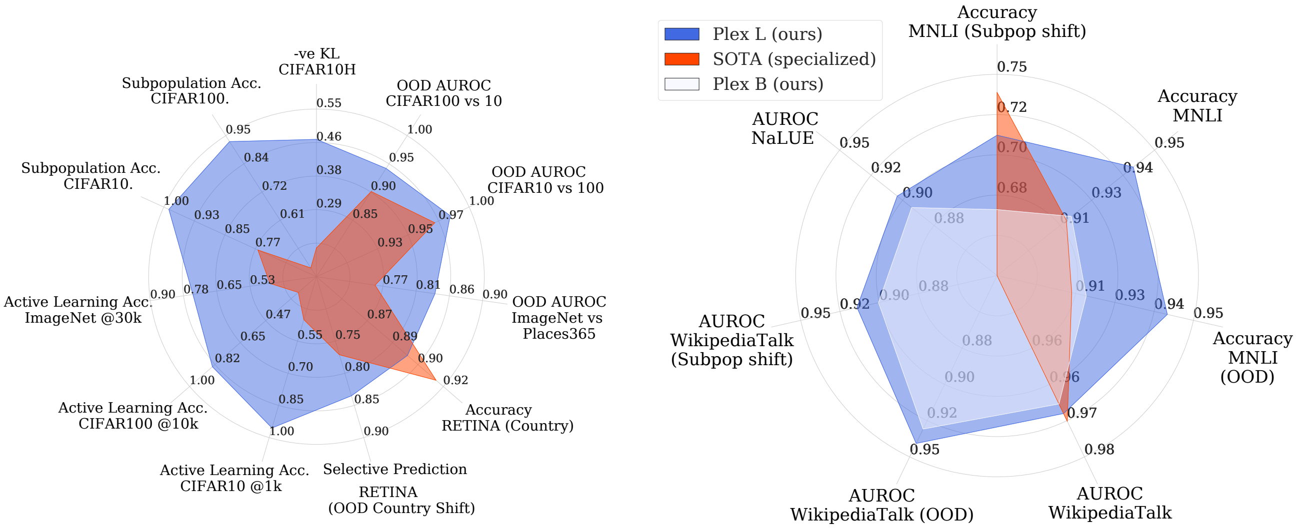


Figure 3. ViT-Plex (**left**) and T5-Plex (**right**) evaluated on a highlighted set of reliability tasks. We also display the state-of-the-art for each task. ViT-Plex and T5-Plex significantly improve state-of-the-art across multiple tasks. Importantly, Plex unifies reliability performance under one general model for vision and language respectively as opposed to specific techniques for each downstream task.

are effectively tokenized into patches.

- **Model size.** We investigate 3 scales of the model size in ViT Plex (S/32, B/32, L/32) and 3 scales of the model size in T5 Plex (Small, Base, Large).
- **Pretraining dataset size.** For vision, we scale pretraining from ImageNet21K to the JFT web dataset on up to 4B images. This mirrors recent work on scaling vision models (Zhai et al., 2021; Pham et al., 2021). For language, we use the C4 dataset which consists of hundreds of gigabytes of English text scraped from the web (Raffel et al., 2020).
- **Efficient ensembling.** Ensembles and Bayesian neural nets have shown to be very effective for uncertainty and robustness (Ovadia et al., 2019; Dusenberry et al., 2020; Band et al., 2021). To do so scalably, we use BatchEnsemble (BE) (Wen et al., 2020) and experiment with its use on both the attention and feedforward layers or on only the feedforward layer. For faster training, we only apply BatchEnsemble at a select number of later layers, similar to mixture of experts models (Riquelme et al., 2021).
- **Last layer changes.** We experiment with two approaches that modify the model's final layer to improve reliability, given a fixed representation (a.k.a. *deterministic uncertainty quantification* setting (Van Amersfoort et al., 2020)). First, we use Gaussian process (GP) last-layer, which improve distance-awareness of the decision surface by increasing uncertainty far away from the training representations. We use the GP layer implementation proposed by Liu et al. (2020). In addition, pretraining uses increasingly noisier datasets with a large number of output classes, and the ability to model input-dependent label noise becomes more important. We apply the Heteroscedastic (Het) method of Collier et al. (2021).

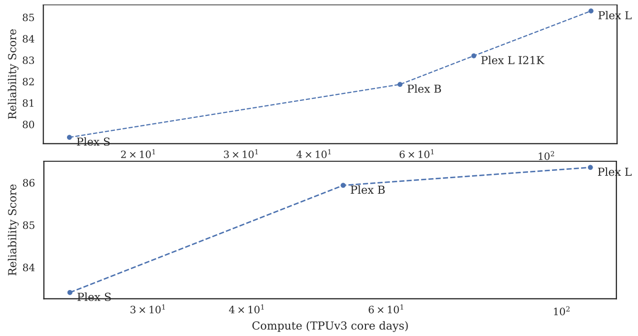


Figure 4. Performance aggregated across 52 vision task metrics. Compute is the total # of training days for a single TPUv3 core.

- **What to apply in pretraining versus finetuning.** We experiment with both pretraining and finetuning for vision models. Due to compute constraints, we exclusively focus on the finetuning-only setting for T5-Plex. That is, T5-Plex models are initialized from the official pretrained T5 checkpoints, and we apply efficient ensembling and last layer changes during finetuning.
- **Few-shot protocol.** As an alternative to logistic regression on the final layer of frozen representations, we experiment with gradient descent over all parameters. We also experiment with a GP or Heteroscedastic last layer.

4 Summary of Results and Scaling Trends

Figure 3 displays our model's overall performance comparing reliability task performance to existing specialized state-of-the-art. Here, we validate several takeaways as we ablate to understand the ingredients behind Plex.

Scaling model size improves reliability. Figure 4 displays ViT Plex of varying model sizes pretrained with JFT; I21K denotes pretraining on ImageNet21K. We compute a reliability score which is an average over all 52 task metrics

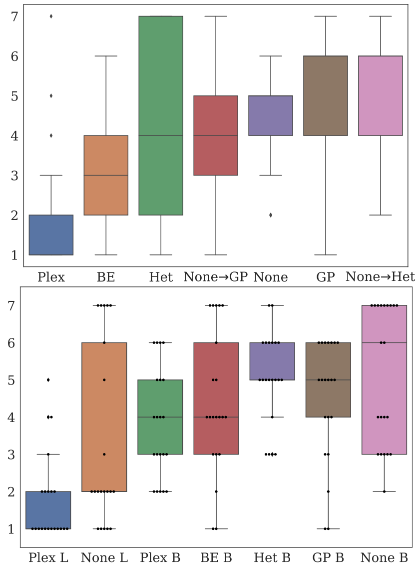


Figure 5. Ranking of method ablations over **(top)** 52 metrics on vision tasks and **(bottom)** 21 metrics on language tasks. Each model has a box plot of rankings (lower is better). Plex's combination of efficient ensembling and last layer changes ranks best on average.

(see Appendix B for details). Classical machine learning theory would suggest that a larger model translates to more overfitting and might therefore be less reliable as it may be overconfident and less robust. However, we find that scale improves overall performance across tasks.

Scaling pretraining dataset size improves reliability. The models tracing the Pareto frontier in Figure 4 use JFT for pretraining. Plex L/32 and even B/32 with JFT performs better than Plex L/32 with ImageNet21K. This suggests that larger and diverse pretraining data is better for reliability.

BatchEnsemble improves pretraining. For vision, we run ablations at the fixed setting of L/32 pretrained with JFT, and we use both B and L sizes for text, which are highly competitive settings. Figure 5 displays the ranking across tasks for each model. Methods are applied either during both pretraining and finetuning, or only during finetuning given a pretrained model checkpoint ("BE→Het" means pretraining with BE and finetuning with Het on top). All the methods displayed improve over a baseline without ensembling or last layer changes (None). BatchEnsemble is consistently the best for pretraining. For T5 on text, Plex Large outperforms Plex Base, showing the benefits of scale.

Last-layer methods improve finetuning. The best ranked models for the vision and language tasks use all of Plex's ingredients: Het on top of a pretrained BE for vision and GP on top of a BE for language. In particular, for T5-Plex ablations, BE+GP and BE tend to have the strongest performance. From more detailed per-dataset analysis in Appendix E, BE+GP and BE perform well on MNLI and NaLUE with BE+GP performing slightly better; notably, they outperform even an expensive deep ensemble baseline which also performs well on MNLI and NaLUE. BE+GP outperforms None on Toxic Comments while a Monte Carlo Dropout baseline performs best on that task. T5-Plex L also outperforms T5-Plex B, which indicates the benefit of scale not only in Figure 4's normalized average score but in their average ranking.

References

Abraham, B. and Ledolter, J. *Statistical methods for forecasting*, volume 179. Wiley Online Library, 1983.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Avin, S., Belfield, H., Brundage, M., Krueger, G., Wang, J., Weller, A., Anderljung, M., Krawczuk, I., Krueger, D., Lebensold, J., et al. Filling gaps in trustworthy development of ai. *Science*, 374(6573):1327–1329, 2021.

Band, N., Rudner, T. G. J., Feng, Q., Filos, A., Nado, Z., Dusenberry, M. W., Jerfel, G., Tran, D., and Gal, Y. Benchmarking bayesian deep learning on diabetic retinopathy detection tasks. In *NeurIPS Datasets and Benchmarks Track*, 2021.

Barbu, A., Mayo, D., Alverio, J., Luo, W., Wang, C., Gutfreund, D., Tenenbaum, J., and Katz, B. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *NeurIPS*, 32, 2019.

Barlow, R. E. and Proschan, F. Statistical theory of reliability and life testing: probability models. Technical report, Florida State Univ Tallahassee, 1975.

Beyer, L., Hénaff, O. J., Kolesnikov, A., Zhai, X., and Oord, A. v. d. Are we done with ImageNet? *arXiv preprint arXiv:2006.07159*, 2020.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Borkan, D., Dixon, L., Sorensen, J., Thain, N., and Vasserman, L. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*, pp. 491–500, 2019.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *NeurIPS*, 2020.

Collier, M., Mustafa, B., Kokiopoulou, E., Jenatton, R., and Berent, J. A simple probabilistic method for deep classification under input-dependent label noise. *arXiv preprint arXiv:2003.06778*, 2020.

Collier, M., Mustafa, B., Kokiopoulou, E., Jenatton, R., and Berent, J. Correlated input-dependent label noise in large-scale image classification. In *CVPR*, pp. 1551–1560, 2021.

D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M. D., et al. Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*, 2020.

Dawid, A. P. The well-calibrated Bayesian. *Journal of the American Statistical Association*, 1982.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Dusenberry, M., Jerfel, G., Wen, Y., Ma, Y., Snoek, J., Heller, K., Lakshminarayanan, B., and Tran, D. Efficient and scalable Bayesian neural nets with rank-1 factors. In *ICML*, 2020.

Everitt, T., Lea, G., and Hutter, M. AGI safety literature review. *arXiv preprint arXiv:1805.01109*, 2018.

Fort, S., Ren, J., and Lakshminarayanan, B. Exploring the limits of out-of-distribution detection. In *NeurIPS*, 2021.

Ghahramani, Z. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452–459, 2015.

Gneiting, T. and Raftery, A. E. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477):359–378, March 2007. ISSN 0162-1459. doi: 10.1198/016214506000001437.

Gustafsson, F. K., Danelljan, M., and Schön, T. B. Evaluating scalable Bayesian deep learning methods for robust computer vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020.

Hendrycks, D. and Dietterich, T. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *International Conference on Learning Representations*, 2019.

Hendrycks, D., Lee, K., and Mazeika, M. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, 2019a.

Hendrycks, D., Mazeika, M., Kadavath, S., and Song, D. Using self-supervised learning can improve model robustness and uncertainty. *arXiv preprint arXiv:1906.12340*, 2019b.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, 2021a.

Hendrycks, D., Carlini, N., Schulman, J., and Steinhardt, J. Unsolved problems in ML safety. *arXiv preprint arXiv:2109.13916*, 2021b.

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. Natural adversarial examples. In *CVPR*, pp. 15262–15271, 2021c.

Kendall, A. and Gal, Y. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

Kivlichan, I. D., Lin, Z., Liu, J., and Vasserman, L. Measuring and improving model-moderator collaboration using uncertainty estimation. *arXiv preprint arXiv:2107.04212*, 2021.

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., and Houlsby, N. Big transfer (bit): General visual representation learning. In *ECCV*, 2020.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. *arXiv preprint*, 2009.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, volume 30, 2017.

Larson, S., Mahendran, A., Peper, J. J., Clarke, C., Lee, A., Hill, P., Kummerfeld, J. K., Leach, K., Laurenzano, M. A., Tang, L., et al. An evaluation dataset for intent classification and out-of-scope prediction. *arXiv preprint arXiv:1909.02027*, 2019.

Liu, J., Lin, Z., Padhy, S., Tran, D., Bedrax Weiss, T., and Lakshminarayanan, B. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *NeurIPS*, 2020.

Liu, J. Z., Padhy, S., Ren, J., Lin, Z., Wen, Y., Jerfel, G., Nado, Z., Snoek, J., Tran, D., and Lakshminarayanan, B. A simple approach to improve single-model deep uncertainty via distance-awareness. *arXiv preprint arXiv:2205.00403*, 2022.

Liu, X., Eshghi, A., Swietojanski, P., and Rieser, V. Benchmarking natural language understanding services for building conversational agents. In *Increasing Naturalness and Flexibility in Spoken Dialogue Interaction*, pp. 165–183. Springer, 2021.

McCoy, T., Pavlick, E., and Linzen, T. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334. URL `https://aclanthology.org/P19-1334`.

Minderer, M., Djolonga, J., Romijnders, R., Hubis, F., Zhai, X., Houlsby, N., Tran, D., and Lucic, M. Revisiting the calibration of modern neural networks. *NeurIPS*, 2021.

Nado, Z., Band, N., Collier, M., Djolonga, J., Dusenberry, M. W., Farquhar, S., Feng, Q., Filos, A., Havasi, M., Jenatton, R., et al. Uncertainty baselines: Benchmarks for uncertainty & robustness in deep learning. *arXiv preprint arXiv:2106.04015*, 2021.

Naeini, M. P., Cooper, G., and Hauskrecht, M. Obtaining well calibrated probabilities using Bayesian binning. In *AAAI*, 2015.

O'Connor, P. and Kleyner, A. *Practical reliability engineering*. John Wiley & Sons, 2012.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019.

Parmigiani, G. and Inoue, L. *Decision theory: principles and approaches*. Wiley series in probability and statistics. John Wiley & Sons, Chichester, West Sussex, U.K. ; [Hoboken, N.J.], 2009. ISBN 978-0-471-49657-1. OCLC: ocn276340596.

Peterson, J. C., Battleday, R. M., Griffiths, T. L., and Russakovsky, O. Human uncertainty makes classification more robust. In *ICCV*, 2019.

Pham, H., Dai, Z., Ghiasi, G., Liu, H., Yu, A. W., Luong, M.-T., Tan, M., and Le, Q. V. Combined scaling for zero-shot transfer learning. *arXiv preprint arXiv:2111.10050*, 2021.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In *ICLR*, 2017.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do ImageNet classifiers generalize to ImageNet? In *ICML*, pp. 5389–5400. PMLR, 2019.

Ren, J., Fort, S., Liu, J., Roy, A. G., Padhy, S., and Lakshminarayanan, B. A simple fix to mahalanobis distance for improving near-ood detection. *arXiv preprint arXiv:2106.09022*, 2021.

Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Susano Pinto, A., Keysers, D., and Houlsby, N. Scaling vision with sparse mixture of experts. *NeurIPS*, 34, 2021.

Russell, S., Dewey, D., and Tegmark, M. Research priorities for robust and beneficial artificial intelligence. *Ai Magazine*, 36(4):105–114, 2015.

Settles, B. Active learning literature survey. *preprint*, 2009.

Shankar, V., Dave, A., Roelofs, R., Ramanan, D., Recht, B., and Schmidt, L. Do image classifiers generalize across time? In *ICCV*, 2021.

Sugiyama, M. and Kawanabe, M. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.

Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

Thrun, S. Lifelong learning algorithms. In *Learning to learn*, pp. 181–209. Springer, 1998.

Tran, D., Hoffman, M. W., Moore, D., Suter, C., Vasudevan, S., and Radul, A. Simple, distributed, and accelerated probabilistic programming. *Advances in Neural Information Processing Systems*, 31, 2018.

Tran, D., Snoek, J., and Lakshminarayanan, B. Practical uncertainty estimation and out-of-distribution robustness in deep learning. *NeurIPS tutorial*, 2020.

Van Amersfoort, J., Smith, L., Teh, Y. W., and Gal, Y. Uncertainty estimation using a single deep deterministic neural network. In *ICML*, 2020.

Wen, Y., Tran, D., and Ba, J. BatchEnsemble: an Alternative Approach to Efficient Ensemble and Lifelong Learning. In *ICLR*, 2020.

Williams, A., Nangia, N., and Bowman, S. R. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

Wulczyn, E., Thain, N., and Dixon, L. Ex machina: Personal attacks seen at scale. In *WWW*, 2017.

Zhai, X., Kolesnikov, A., Houlsby, N., and Beyer, L. Scaling vision transformers. *arXiv preprint arXiv:2106.04560*, 2021.

Zhang, J.-G., Hashimoto, K., Wan, Y., Liu, Y., Xiong, C., and Yu, P. S. Are pretrained transformers robust in intent classification? a missing ingredient in evaluation of out-of-scope intent detection. *arXiv preprint arXiv:2106.04564*, 2021.

A Setup and Downstream Datasets

Figure 6 describes our overall experimental setup. Next, we describe the individual datasets for each modality.

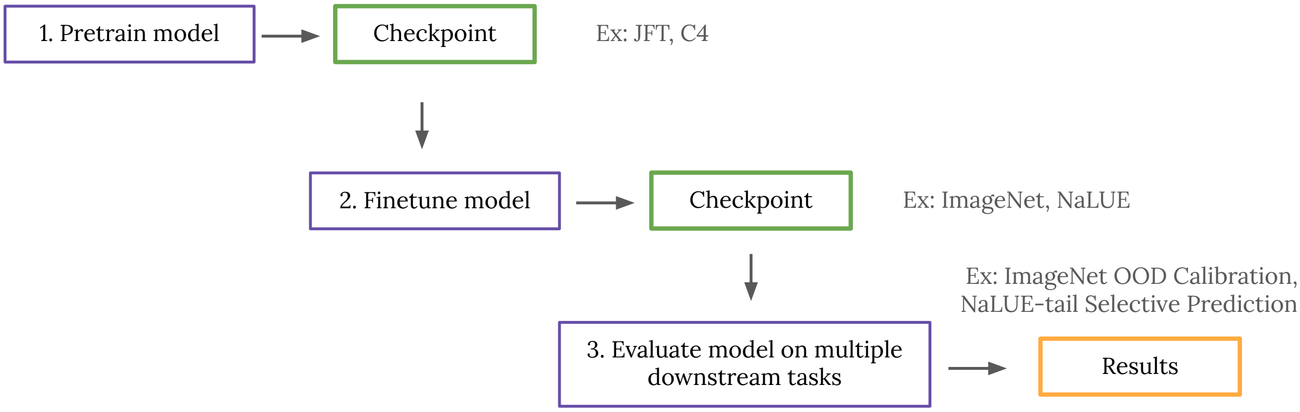


Figure 6. An overview of the model and task pipeline. A choice of pretrained model is trained; given the pretrained model's checkpoint, we apply a variety of methods for finetuning; finally, given the finetuned checkpoint, we evaluate the model on downstream metrics.

Images. We use 11 datasets for training and in-distribution evaluation, and 17 datasets for out-of-distribution evaluation:

- CIFAR-10 has a training set of 50,000 examples and a test set of 10,000 examples (Krizhevsky et al., 2009). Following Dosovitskiy et al. (2020), we use 99% of the training set for training and 1% for validation.
- CIFAR-100 has a training set of 50,000 examples and a test set of 10,000 examples. Following Dosovitskiy et al. (2020), we use 99% of the training set for training and 1% for validation.
- ImageNet 1K has a training set of roughly 1.2 million examples and a test set of 50,000 examples (Deng et al., 2009). Following Dosovitskiy et al. (2020), we use 98% of the training set for training and 2% for validation.
- EyePACS is a dataset for diabetic retinopathy. We chose it as an example of a difficult transfer task, far away from the distribution of natural images on the web seen during pretraining. There are two common constructions of EyePACS, following the RETINA benchmark (Band et al., 2021): a Severity split results in 28,253 examples in the training set and 49,543 in the test sets; and a Country split results in 35,126 examples in the training set and 45,559 examples in the test sets.
- We use an assortment of datasets each with less than 15,000 examples: Caltech-UCSD Birds 200, Caltech 101, Cars196, Colorectal histology, Describable Textures Dataset, Oxford-IIIT pet, UC Merced.

With these training datasets, we attempt to cover multiple types of out-of-distribution shifts:

- Covariate shift refers to scenarios where the distribution of inputs changes while the conditional distribution of outputs is unchanged (Sugiyama & Kawanabe, 2012). For example, the training set may include natural cat images and the new input is a cat image after applying synthetic image corruptions.
  - CIFAR-10: CIFAR-10-C (Hendrycks & Dietterich, 2019).
  - CIFAR-100: CIFAR-100-C (Hendrycks & Dietterich, 2019).
  - ImageNet-1K: ImageNet-A (Hendrycks et al., 2021c), ImageNet-C (Hendrycks & Dietterich, 2019), ImageNetV2 (Recht et al., 2019), ImageNet-Vid-Robust, YTTB Robust (Shankar et al., 2021), ObjectNet (Barbu et al., 2019), and ImageNet-R (Hendrycks et al., 2021a).
  - EyePACS: RETINA's Country Shift dataset (Band et al., 2021).
- Semantic shift refers to scenarios where the test inputs may belong to different classes than the training classes. For example, the training set may consist only of cat images and the new input is a dog image.
  - CIFAR-10: CIFAR-100, SVHN.
  - CIFAR-100: CIFAR-10, SVHN.

  - ImageNet 1K: Places365.
  - EyePACS: RETINA's Severity Shift dataset (Band et al., 2021).

- Data uncertainty refers to scenarios where there are multiple labels per input representing an underlying soft label (probability distribution) that is ground truth. For example, the dataset may consist of images each with multiple ratings accounting for human uncertainty around the correct label. We use CIFAR-10H dataset (Peterson et al., 2019) which captures human uncertainty over labels for CIFAR-10 dataset. We construct a similar variant for ImageNet 1K by building on ImageNet ReaL (Beyer et al., 2020) where we use the raw human ratings to construct soft label targets.

Text. We consider real-world scenarios that are known to deploy machine learning models for decision making: natural language inference (NLI), toxic comments detection on online forums, and conversational language understanding (CLU).

- For NLI, we consider the Multi-Genre Natural Language Inference (MNLI) corpus which is consistent of 433k sentence pairs from a diverse collection of genres (fiction, government report, news magazine articles, etc) (Williams et al., 2017).
- For toxic comments detection, we consider the WikipediaTalk corpus (Wulczyn et al., 2017) which is composed of 200k English Wikipedia talk page comments between Wikipedia editors across the world.
- For CLU, a large-scale corpus for evaluating uncertainty quantification in intent understanding is lacking. We propose a new dataset, Natural Language understanding Uncertainty Evaluation (NaLUE) that is a relabelled and aggregated version of three large NLU corpuses CLINC150 (Larson et al., 2019), Banks77 (Zhang et al., 2021) and HWU64 (Liu et al., 2021) contains 50k+ utterances spanning 18 verticals, 77 domains, and 260 intents. For this task, the model needs to map each utterance to a 3-token sequence of (vertical name, domain name, intent name).

Natural language is diverse, fast evolving, and rich in long-tail linguistic phenomena. Therefore out-of-distribution and long-tail examples are pervasive in the real-world deployment environment. To gain a full understanding of method performance in these situations, we also design out-of-domain challenge sets for each dataset. Specifically, for covariate and semantic shift, we use:

- the MNLI-mismatched (Williams et al., 2017) data as the OOD set for NLI, which contains sentence pairs from 5 genres that are distinct from those in MNLI training data.
- the CivilComments corpus (Borkan et al., 2019) as the OOD set for toxic comment prediction, which consists of one million public comments appearing on approximately 50 English-language news sites across the world.

For subpopulation shift under long-tail groups, we use:

- HANS (McCoy et al., 2019) eval datasets for NLI, which contains template-generated examples attacking the surface-level heuristics that the neural models are found to rely on when predicting entailment relationships.
- CivilCommentsIdentity (Borkan et al., 2019) for toxic comments, which is a subset of CivilComments that has explicit mention of social identities (e.g., muslim, LGBTQ, etc) that the model are often found to generate mispredictions.
- NaLUE-tail dataset for CLU, which is a subset of NaLUE corresponding to utterances from 28 low-frequency intents categories. Together, the datasets constitute 3 finetuning and 5 out-of-distribution or long-tail recognition tasks.

B Details of Overall Reliability Score

In Figure 4, we aggregate all task metrics under a single scalar between 0 and 100. In order to do this, we must normalize all metrics to be between 0 and 100; we then compute an unweighted average. Most metrics are already bounded between 0 and 100: for example, accuracy, expected calibration error (we do 100 − ECE so higher is better), calibration AUC, and AUROC. The one exception are scoring rules such as log-loss and Brier score. Because the output distributions are discrete, log-loss has a lower bound of 0 and an upper bound given by the highest entropy distribution (uniform). Therefore we rescale scoring rule values based on their lower and upper bounds so that they're now between 0 and 100 and so that higher is better.

C Details of Plex ingredients

In this work, we focus on two domains: images and text. For images, we use a base architecture of Vision Transformer that performs image classification (Dosovitskiy et al., 2020). For text, we use T5 which uses an encoder-decoder architecture to treat text problems as text input and text output (Raffel et al., 2020). On top of these architectures, we experiment with the following methods.

BatchEnsemble (BE). BatchEnsembles (Wen et al., 2020) approximate deep ensembles (Lakshminarayanan et al., 2017), but reduce their computational and memory costs by sharing weights across the ensemble members. The weight matrix $\boldsymbol{W}_i$ of any given ensemble member i is written as the Hadamard product of a shared weight matrix $\boldsymbol{W}_0$ and a local rank-1 matrix $r_i s_i^\top$:

$$\boldsymbol{W}_i = \boldsymbol{W}_0 \circ r_i s_i^\top . \tag{1}$$

The vectors r and s are commonly referred to as fast weights.

Unless otherwise stated, Plex applies BE to all layers in the last 2 residual blocks of the network. This idea follows work for mixture of experts (Riquelme et al., 2021).

Spectral-normalized Neural Gaussian Process (SNGP). Unlike ensemble approaches, SNGP proposed by Liu et al. (2020) focuses on improving the uncertainty quality of a neural network given a fixed representation (a.k.a. *deterministic uncertainty quantification* setting (Van Amersfoort et al., 2020). When applied to a DNN without pretraining, SNGP enhances the DNN uncertainty property by applying spectral normalization to the hidden weights, and replaces the output layer from a dense layer to a random-feature Gaussian process (GP) layer. That is, given hidden representations $h(\boldsymbol{x})$, the GP layer enables scalable computation of a GP posterior by applying a random feature approximation ϕ to the predictive function and then a Laplace approximation to the predictive variance:

$$\begin{aligned} g(\boldsymbol{x}) &\sim N(\text{logit}(\boldsymbol{x}), \text{var}(\boldsymbol{x})) \\ \text{logit}(\boldsymbol{x}) &= \phi(x)^\top \beta, \quad \text{where} \\ \phi(\boldsymbol{x}) &= \cos(\boldsymbol{W} h(\boldsymbol{x}) + \boldsymbol{b}) \\ \text{var}(x) &= \phi(\boldsymbol{x})^\top (I + \Phi^T \Phi)^{-1} \phi(\boldsymbol{x}) \end{aligned}$$

where $(\boldsymbol{W}, \boldsymbol{b})$ are frozen random weights of the random feature embedding $\phi(\boldsymbol{x}) = \cos(\boldsymbol{W} h(\boldsymbol{x}) + \boldsymbol{b})$, and $\Phi^\top \Phi = \sum_i \phi(\boldsymbol{x}_i) \phi(\boldsymbol{x}_i)^\top$ is the covariance of the random feature embedding estimated using the training data.

Liu et al. (2020; 2022) show that this combined technique improves the model's awareness of the semantic distance between the test and train examples on the data manifold, leading to improved performance in calibration and out-of-domain detection. When applied to a large pretrained DNN, we find it sufficient to only use the last-layer Gaussian process (i.e., omit the spectral normalization regularization), as the pre-trained embedding has already provided a semantic-distance-aware representation of the data.

Heteroscedastic last layer (Het). Heteroscedastic last layers are designed to model input-dependent label noise/data uncertainty (a.k.a. aleatoric uncertainty (Kendall & Gal, 2017)) that is present in the data. We use the Heteroscedastic (Het) last layer introduced by Collier et al. (2020; 2021) who place a multivariate Gaussian distribution over the logits in a standard DNN classifier. A low-rank approximation to the $K \times K$ covariance matrix (K = number of classes/outputs) is made when K is large and (Collier et al., 2021) further develop a parameter efficient version of the method with parameterization inspired by BE to enable scaling to tens of thousands of classes.

Naming of different methods We apply these modifications either during both pretraining and finetuning, or only during finetuning given a pretrained model checkpoint. None refers to the baseline without ensembling or last layer changes. "None→GP" means standard pretraining (without any modifications) and just applying GP layer during finetuning. "BE" means using BE during both pretraining and finetuning. "BE→Het" means pretraining with BE and finetuning with Het on top.

D Related Work

Prior work has investigated a variety of approaches to improve narrower definitions of reliability. From the literature, several overarching dimensions arise (Tran et al., 2020)—such as the importance of model and data size (e.g. pretraining); model inductive biases (e.g. architecture and data augmentation); and the combination of multiple models (e.g. ensembles and Bayesian neural networks). There is not yet an understanding of how these dimensions interact (and within current literature, it is no surprise that there are contradicting messages) and which of these dimensions provide complementary benefits. We investigate how each of these dimensions improve reliability and how they can be "composed" to maximize performance.

Modern AI is trending towards training a single large model on a large data set, known as pretraining, and then applying the model to a wide variety of related downstream tasks (Radford et al., 2021; Brown et al., 2020; Thoppilan et al., 2022; Kolesnikov et al., 2020). This often improves over task-specific state-of-the-art in predictive performance, with many considering such large scale models to represent a "paradigm shift" in ML (Bommasani et al., 2021). Large-scale pre-trained models have also significantly improved state-of-the-art on narrower tasks such as accuracy and calibration under covariate shift, see (Minderer et al., 2021; Hendrycks et al., 2019a;b) (as well as (Bommasani et al., 2021, Section 4.8) for additional references) and open set recognition (cf. (Fort et al., 2021; Ren et al., 2021)). Given these initial promising results, we use large-scale pre-trained models as a building block for investigating reliability. However, large models can be compute intensive, which warrants revisiting existing recipes; for instance, vanilla deep ensembles, which work well in previous benchmarks (Ovadia et al., 2019; Gustafsson et al., 2020; Band et al., 2021), might be computationally expensive. Hence, we focus on scalable modifications to large models such as efficient ensembles and last-layer variants, detailed in Appendix C.

E Summarization of Language Results

We first compare the performance across types of uncertainty methods, fixing the architecture size to T5-base. We compare performances in prediction, uncertainty calibration, and human-model collaboration, across all datasets (MNLI, NaLUE and Toxic Comments) and all splits (In-domain, OOD, and tail-population). Table 2 reports the full results, and Figure 7 summarizes the rankings of uncertainty methods under each type of population shift (in-domain v.s. OOD v.s. tail-population). Among all methods, DE+GP, Plex (i.e., BE+GP), BE, and MC Dropout tend to have the strongest performance. In particular, DE+GP almost always dominates the other methods on MNLI and NaLUE, and remains competitive in the case of label imbalance (i.e., Toxic Comments). However, DE+GP is an expensive method that costs x10 more in memory and compute and therefore is not competitive in scale (a more thorough analysis is in **??**). On the other hand, among the more efficient, single-model methods, BE and Plex perform well on MNLI and NaLUE (notably, outperform the most expensive DE), while MCD stands out in the Toxic Comments. The above observations suggest that, when the training example has a simple distribution, quantifying output-layer uncertainty alone is sufficient to attain strong performance. However, when there are pathologies in the data distribution (e.g., extreme label imbalance), quantifying the uncertainty within the model's intermediate representations (e.g., via some form of perturbation like BE) becomes important.

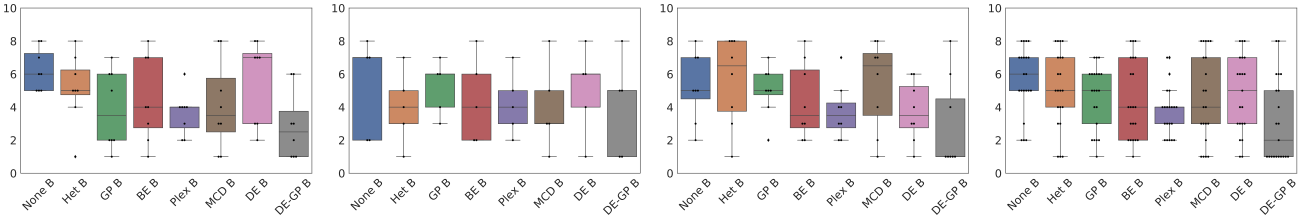
We then investigate how a model's uncertainty performance is impacted by the architecture size. For model size scaling, we evaluate Plex, None, and MC Dropout, the three best-performing and efficient methods in the previous study. We evaluate the performance of each method under three progressively larger architectures: T5 S, T5 B, and T5 L, and observe how the behavior changes across the method and with respect to the architecture size. Table 2 reports the full results, and Figure 8 summarizes the rankings of uncertainty methods organized by the sizes of the architecture. As shown, comparing across architecture sizes, we see a larger architecture almost always leads to stronger performance in collaborative performance. This trend remains largely consistent even when out-of-distribution.

Task	Split	Score	None B	Het B	GP B	BE B	Plex B	MCD B	DE B	DE-GP B
MNLI	In-domain	calibration	0.381	0.384	0.372	0.401	0.388	0.416	0.383	0.4
		generalization	0.938	0.949	0.944	0.949	0.948	0.946	0.938	0.95
		select. pred.	0.961	0.971	0.968	0.973	0.973	0.973	0.961	0.975
	OOD	calibration	0.391	0.401	0.393	0.413	0.394	0.41	0.388	0.416
		generalization	0.937	0.948	0.941	0.949	0.948	0.946	0.938	0.95
		select. pred.	0.959	0.971	0.966	0.973	0.973	0.972	0.96	0.975
	Subpopulation	calibration	0.451	0.434	0.454	0.443	0.401	0.474	0.443	0.418
		generalization	0.749	0.766	0.762	0.791	0.788	0.739	0.764	0.798
		select. pred.	0.811	0.824	0.842	0.87	0.871	0.831	0.826	0.885
NaLUE	In-domain	calibration	0.498	0.484	0.512	0.464	0.486	0.471	0.487	0.494
		generalization	0.939	0.932	0.94	0.935	0.94	0.938	0.942	0.944
		select. pred.	0.936	0.935	0.932	0.938	0.938	0.932	0.937	0.938
	OOS, Near	detection	0.706	0.673	0.719	0.768	0.716	0.766	0.721	0.771
	OOS, Standard	detection	0.964	0.957	0.992	0.998	0.991	0.994	0.973	0.998
	Subpopulation	calibration	0.518	0.511	0.553	0.514	0.519	0.466	0.513	0.514
		generalization	0.866	0.846	0.869	0.87	0.873	0.858	0.871	0.882
		select. pred.	0.862	0.82	0.828	0.845	0.856	0.829	0.861	0.851
Toxic Comments	In-domain	calibration	0.459	0.462	0.46	0.461	0.471	0.442	0.459	0.465
		generalization	0.888	0.89	0.899	0.889	0.895	0.904	0.885	0.892
		select. pred.	0.936	0.938	0.94	0.938	0.94	0.941	0.936	0.939
	OOD	calibration	0.425	0.427	0.438	0.423	0.447	0.413	0.426	0.421
		generalization	0.82	0.817	0.818	0.81	0.816	0.831	0.817	0.818
		select. pred.	0.86	0.857	0.855	0.85	0.855	0.862	0.86	0.852
	Subpopulation	calibration	0.415	0.405	0.421	0.412	0.428	0.405	0.416	0.4
		generalization	0.806	0.803	0.804	0.795	0.801	0.814	0.801	0.803
		select. pred.	0.831	0.828	0.828	0.821	0.826	0.835	0.83	0.823

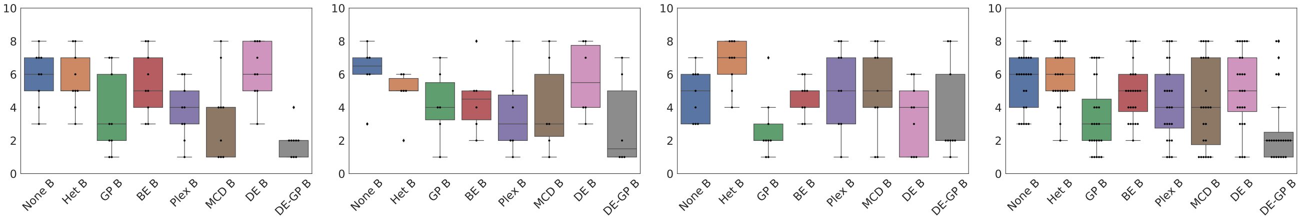
Table 1. Comparison of method performance between uncertainty methods. Black: Best. Dark Grey: Second. Light Grey: Third.

Task	Split	Score	None S	MCD S	Plex S	None B	MCD B	Plex B	None L	MCD L	Plex L
MNLI	In-domain	calibration	0.399	0.406	0.364	0.381	0.416	0.388	0.39	0.404	0.394
		generalization	0.924	0.927	0.913	0.938	0.946	0.948	0.963	0.964	0.965
		select. pred.	0.953	0.959	0.942	0.961	0.973	0.973	0.982	0.985	0.985
	OOD	calibration	0.398	0.396	0.367	0.391	0.41	0.394	0.418	0.411	0.406
		generalization	0.924	0.93	0.911	0.937	0.946	0.948	0.963	0.965	0.967
		select. pred.	0.953	0.96	0.94	0.959	0.972	0.973	0.983	0.986	0.987
	Subpopulation	calibration	0.555	0.56	0.514	0.451	0.474	0.401	0.417	0.45	0.447
		generalization	0.648	0.619	0.59	0.749	0.739	0.788	0.817	0.807	0.803
		select. pred.	0.747	0.717	0.677	0.811	0.831	0.871	0.875	0.896	0.876
NaLUE	In-domain	calibration	0.507	0.48	0.498	0.498	0.471	0.486	0.486	0.453	0.496
		generalization	0.942	0.932	0.937	0.939	0.938	0.94	0.931	0.928	0.944
		select. pred.	0.937	0.926	0.929	0.936	0.932	0.938	0.93	0.922	0.935
	OOS, Near	detection	0.71	0.756	0.689	0.706	0.766	0.716	0.692	0.733	0.781
	OOS, Standard	detection	0.968	0.992	0.999	0.964	0.994	0.991	0.956	0.991	0.991
	Subpopulation	calibration	0.528	0.462	0.499	0.518	0.466	0.519	0.518	0.466	0.492
		generalization	0.878	0.854	0.851	0.866	0.858	0.873	0.843	0.83	0.871
		select. pred.	0.864	0.836	0.84	0.862	0.829	0.856	0.835	0.801	0.835
Toxic Comments	In-domain	calibration	0.455	0.445	0.478	0.459	0.442	0.471	0.448	0.451	0.436
		generalization	0.879	0.898	0.863	0.888	0.904	0.895	0.886	0.906	0.89
		select. pred.	0.932	0.938	0.919	0.936	0.941	0.94	0.936	0.944	0.942
	OOD	calibration	0.423	0.412	0.425	0.425	0.413	0.447	0.432	0.417	0.459
		generalization	0.81	0.823	0.807	0.82	0.831	0.816	0.823	0.837	0.816
		select. pred.	0.85	0.851	0.838	0.86	0.862	0.855	0.865	0.869	0.863
	Subpopulation	calibration	0.409	0.404	0.412	0.415	0.405	0.428	0.426	0.403	0.46
		generalization	0.795	0.806	0.786	0.806	0.814	0.801	0.809	0.822	0.805
		select. pred.	0.82	0.824	0.803	0.831	0.835	0.826	0.837	0.842	0.838

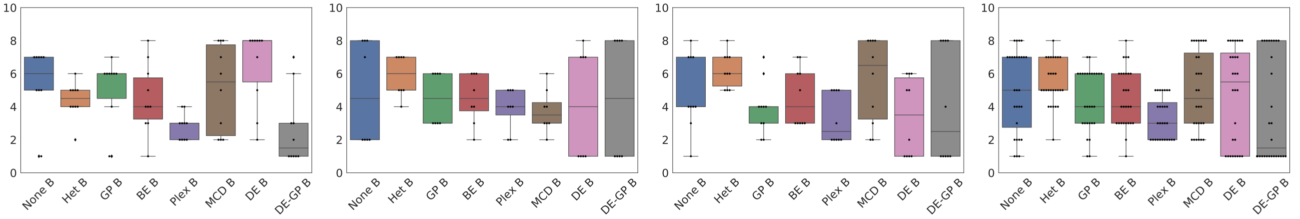
Table 2. Comparison of method performance between architecture sizes. Black: Best. Dark Grey: Second. Light Grey: Third.



(a) Generalization, IND. (b) Generalization., OOD. (c) Generalization, SUB. (d) Generalization, ALL.

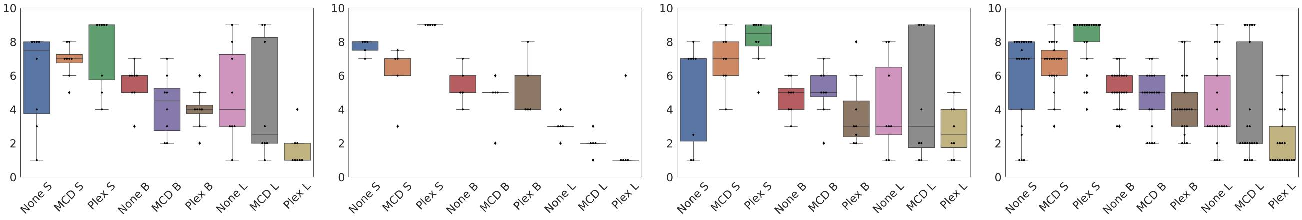


(e) Calibration, IND. (f) Calibration, OOD. (g) Calibration, SUB. (h) Calibration, ALL.

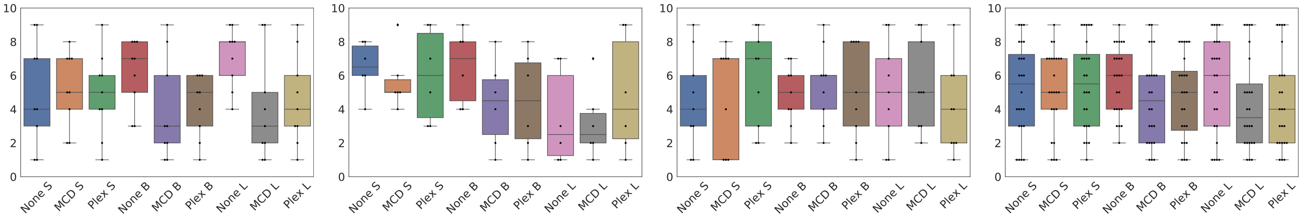


(i) Selective Pred., IND. (j) Selective Pred., OOD. (k) Selective Pred., SUB. (l) Selective Pred., ALL.

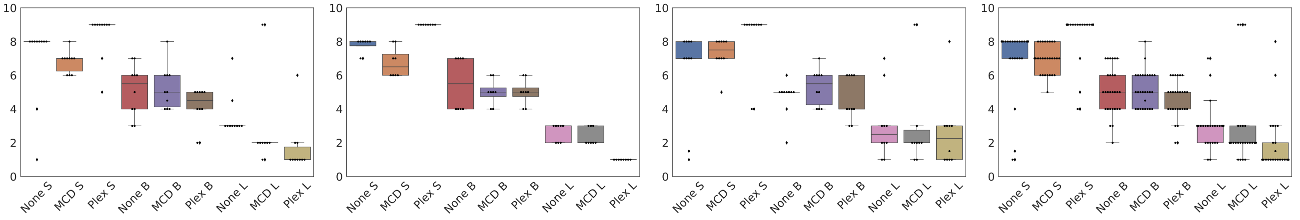
Figure 7. T5-Plex model's ranking comparison between different uncertainty methods and across different evaluation datasets. IND: in-domain. OOD: out-of-domain. SUB: subpopulation shift. ALL: aggregated performance across all datasets.



(a) Generalization, IND. (b) Generalization., OOD. (c) Generalization, SUB. (d) Generalization, ALL.



(e) Calibration, IND. (f) Calibration, OOD. (g) Calibration, SUB. (h) Calibration, ALL.



(i) Selective Pred., IND. (j) Selective Pred., OOD. (k) Selective Pred., SUB. (l) Selective Pred., ALL.

Figure 8. T5-Plex model's ranking comparison between architecture sizes and across different evaluation datasets. IND: in-domain. OOD: out-of-domain. SUB: subpopulation shift. ALL: aggregated performance across all datasets.