Contrastive Classification via Linear Layer Extrapolation

Anonymous EMNLP submission

Abstract

Early-exiting predictions in a deep Transformer network evolve from layer to layer in a somewhat smooth process. This has been exploited in language modeling to improve factuality (Chuang et al., 2023), with the observation that factual associations emerge in later layers. We find a similar process multiway emotion classification, motivating Linear Layer Extrapolation, which finds stable improvements by recasting contrastive inference as linear extrapolation. Experiments across multiple models and emotion classification datasets find that Linear Layer Extrapolation outperforms standard classification on fine-grained emotion analysis tasks.

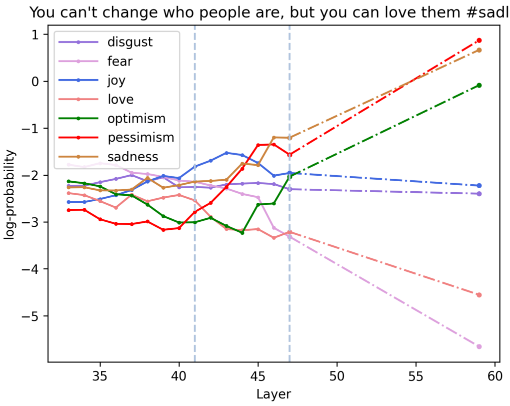


Figure 1: Linearly extrapolating class scores from amateur and expert layers to a nonexistent future layer correctly flips the output from sadness to pessimism.

1 Introduction

Despite the success of large language models on a variety of NLP tasks (Brown et al., 2020; Wei et al., 2022), they still struggle with commonsense reasoning (Fu et al., 2023), often hallucinate incorrect information (Ji et al., 2023) and struggle with factual recall (Wang et al., 2023).

Recently, *contrastive* methods, which maximize differences between a desirable "expert" and undesirable "amateur" model, have been proposed to address these issues (Li et al., 2022; O'Brien and Lewis, 2023; Shi et al., 2023). In particular, decoding by contrasting layers (DoLa) (Chuang et al., 2023) improves factuality by contrasting model outputs against early-exit predictions from intermediate layers of the same model. DoLa works under the premise that later layers encode factuality, and thus late-emerging changes to predictions likely update towards more factual predictions.

Recent work has demonstrated that intermediate layer features can also effectively quantify emotions in text (Sharma et al., 2023). Moreover, early-exiting experiments indicated that distinctions between interrelated emotions also tend to evolve gradually across layers (see Appendix C).

Identifying emotions in text is crucial in NLP for applications ranging from detecting harmful behavior to enhancing conversational agents (Zhang et al., 2023; Barbieri et al., 2020). While many systems often focus on mutually exclusive emotions like joy or sadness, fine-grained emotions like grief and remorse are more nuanced and distinct. Many language-model systems still struggle to classify fine-grained emotions and opinions (Demszky et al., 2020; Zhang et al., 2023). Given this, we explore DoLa-style layer contrast to improve fine-grained emotion classification.

The main contributions of the paper are:

1. Demonstrating the merits of layer contrast on fine-grained emotion classification.
2. Recasting contrastive inference as linear extrapolation to obtain more stable performance with a dynamic contrastive penalty.

2 Related Work

Fine-grained Emotion Analysis: Much work has been done in identifying text sentiment (Rosenthal et al., 2017; Socher et al., 2013) and understanding emotions in social media interactions (Mohammad et al., 2018; Chatterjee et al., 2019; Meaney et al., 2021). However, these efforts often focus on a limited set of emotions. Recent datasets on fine-grained emotion analysis (Demszky et al., 2020; Rashkin et al., 2019) indicate significant scope for improvement in this area.

Early Exiting: Early-exiting predictions are obtained by applying the classification head of a model to the residual stream earlier in the network. These have been used to accelerate inference and dynamically allocate compute on a per-input basis (Teerapittayanon et al., 2016; Elbayad et al., 2020; Schuster et al., 2022).

Contrastive Steering: Contrastive methods optimize the difference in predictions between a favorable “expert" and an unfavorable “amateur," to steer text decoding in language models. (Liu et al., 2021) GeDi (Krause et al., 2020) contrasts between class-specific control codes to improve text-conditioned factuality and emotion control. Coherence boosting (Malkin et al., 2021) provides the language model with only the final k tokens of the prompt to obtain amateur scores, encouraging longer-term coherence over locality. Contrastive Decoding (Li et al., 2022; O'Brien and Lewis, 2023) improves long-form generation and reasoning ability by contrasting between large and small models of the same family. Other works use CD-like methods to reduce model toxicity, surface biases and increase faithfulness to a provided context. (Liu et al., 2021; Yona et al., 2023; Shi et al., 2023)

3 Method

Here, we define the main components of CD and DoLa, along with our proposed method for dynamically selecting contrastive strength. We use early exit probability distributions to choose an amateur layer, contrasting its predictions it against the final layer (the expert). We apply mask candidate classes based on a plausibility constraint to filter out low-probability labels. We experiment with two methods for determining contrastive strength: static β and dynamic β. Details of each component are discussed next.

3.1 Contrastive Classification

We use the formulation of contrastive decoding defined by O'Brien and Lewis (2023). Let p_a be the amateur probability scores and p_e be the expert probability scores. We define the contrastive classification function as:

$$f_{CC}^{(i)} = \begin{cases} (1+\beta)\log p_e^i - \beta \log p_a^i & i \in \mathcal{V}_{valid} \\ -\infty & i \notin \mathcal{V}_{valid} \end{cases}$$

where β is the strength of the contrastive penalty and $\mathcal{V}_{valid}$ is the adaptive plausibility constraint (Li et al., 2022) which defines the set of candidate classes on which contrastive action is applied. Let p_e^c be the expert probability for class $c \in C$. Then $\mathcal{V}_{valid}$ is defined as:

$$\mathcal{V}_{valid} = \{\, c \in C,\, p_e^c \geq \alpha \max_{c \in C} p_e^c \,\}$$

α here is a hyperparameter that gates labels by the scores assigned to them by the expert, protecting against instabilities when dividing the scores of two low-probability candidates admitting only high-probability labels. $\arg\max_i (f_{CC}^{(i)})$ is taken as the predicted label.

3.2 Dynamic premature layer selection

The central challenge with inference-time contrastive methods is the selection of a good amateur model. The model must be similar enough to the expert to model its error distribution, but not so powerful that desirable behavior is penalized.

Contrasting against early-exiting layers provides many potential amateurs to choose from. DoLa selects the “amateur" from a pre-validated set of earlier layers, selecting the one with the most different early-exit token distribution from the final predictions, as measured by Jensen-Shannon Divergence. In short, the amateur layer ℓ_a is chosen as follows:

$$\ell_a = \underset{\ell \in L_{valid}}{\arg\max}\, d(\mathcal{P}(\ell), \mathcal{P}(\ell_{final}))$$

where L_{valid} is the pre-validated set of layers, $\mathcal{P}$ maps a latent layer to its early-exited softmax distribution, and d is some divergence metric between two probability distributions.

The original paper uses Jensen-Shannon Divergence (JSD) for d, but we find slightly better performance with cosine distance.

3.3 Linear Layer Extrapolation

Consider the classification of a single sample x to $c \in \mathcal{C}$, where $\mathcal{C} := \{1, 2, \cdots, |\mathcal{C}|\}$. For this sample, let $f_c(i)$ be the un-normalized score assigned by the model to class c by early-exiting at layer i. f_c is defined over the discrete space $\mathcal{C}$.

Let ℓ_a be the index of the selected early-exit amateur layer, $\ell_f > \ell_a$ be the index of the final model layer, and ℓ_t be the desired post-final layer to be linearly approximated. Note that ℓ_t need not be discrete.

Now let $\hat{f}_c$ be the linear function passing through $(\ell_a, f(\ell_a))$ and $(\ell_f, f(\ell_f))$.

$$\hat{f}_c(\ell) = f(\ell_f) + \left(\frac{f(\ell_f) - f(\ell_a)}{\ell_f - \ell_a}\right)(\ell - \ell_f)$$

Now we can compare this extrapolative form against the common form of contrastive decoding in order to solve for the contrastive strength, β.

$\hat{f}_c(\ell_t) = (1+\beta)f(\ell_f) - \beta f(\ell_a)$

Combining the two, we obtain

$$\beta = \frac{\ell_t - \ell_f}{\ell_f - \ell_a} \quad (1)$$

DoLa keeps β fixed, implicitly allowing ℓ_t to vary based as different earlier layers ℓ_a are adaptively chosen. We find more stable performance by fixing ℓ_t and modifying β based on the earlier chosen layer ℓ_a, a process which we refer to as Linear Layer Extrapolation. Choosing an earlier layer will result in a reduced β value, and vice versa.

4 Experimental Setup

4.1 Datasets and Models

In our experiments, we utilize four fine-grained datasets: goEmotions, SuperTweetEval (tweetEmotion, tweetHate), and EmpatheticDialogues. Detailed descriptions of each dataset can be found in Appendix D. We evaluate performance using precision, recall, and F1 scores. Our experiments employ Flan-T5(L, XL) (Chung et al., 2022) and DeBERTa(L, XL) (He et al., 2021). For DeBERTa-xlarge, we fine-tuned after freezing the initial layers (34/48); for DeBERTa-large, we fine-tuned all layers. For Flan-T5, we fine-tuned both large and xlarge variants after freezing the first (14/24) layers. We employed the Adam optimizer (Kingma and Ba, 2014) with learning rates ranging from 1e-6 to 5e-6 for DeBERTa and 1e-4 to 5e-4 for Flan-T5.

4.2 Decoding Hyperparameters:

Amateur layer: For selecting the amateur layer, we use the dynamic amateur layer selection as defined in Section 3.2. We restrict the amateur layer search space to only the finetuned layers. Let $L = \{\ell_k, \ell_{k+1}, \ell_{k+2}, \cdots, \ell_f\}$ be a subset of the finetuned layers, where k is a hyperparameter defining the start of the search space and, ℓ_f is the final layer of the network. In our experiments, we sweep through the values of k starting from the first finetuned layer and pick the one that results in the best performance. Results of the hyperparameter sweep can be found in Appendix A.

Contrastive Strength (β): We experiment with various fixed values of β between 0 to 1, finding that the best β varies over the selection of model and dataset. In general, values outside the range of $(0, 1)$ harmed performance.

Dynamic Contrastive Strength (β): As discussed in Section 3.3, the post-contrast output is equivalent to a linear extrapolation between the amateur and the expert layer for a future layer(ℓ_t). We use that idea to dynamically decide the value of contrastive strength β. We use ℓ_t as a hyperparameter and then calculate β as a function of amateur layer ℓ_a and expert layer ℓ_f, where $t \in (f, f+25)$ in our experiments.

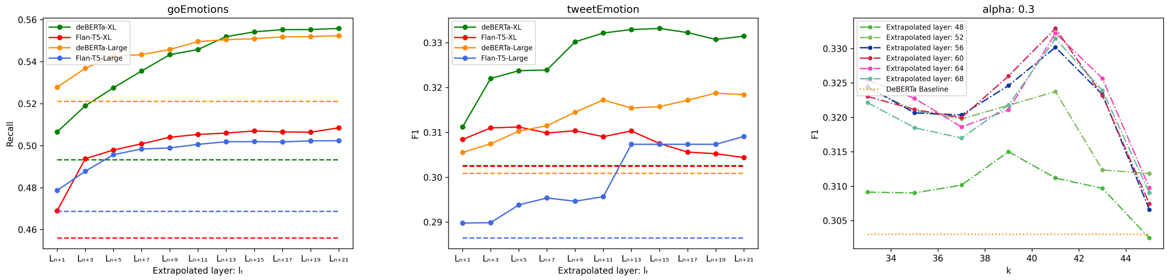
5 Results

Table 1 contains the results of our experiments.

Traditional vs Contrastive Classification: We observe that contrastive classification improves the performance significantly in terms of Recall and F1 score. This trend holds for all models used in our experiments.

β vs Dynamic β: Dynamic β selection tends to improve the overall performance over the static β for F1 and recall scores. Figure 2a shows the trend of recall scores across different models for dynamic β selection on the goEmotions dataset. Figure 2b shows the trend of F1 score across different models against dynamic β for the tweetEmotion dataset. Additionally, we observe that dynamic β is robust to changes in the hyperparameter k, which defines the start of the search space across earlier amateur layers. Figure 3 shows no clear or stable relationship between k and end performance when varying β values. However, switching to linear layer extrapolation creates a constant trend with minor variance as k is varied, a trend that holds for multiple values of extrapolative layer t. This can be interpreted as stabilizing the contrastive method to be more robust to the dynamic choice of amateur layer.

goEmotions: We see a general improvement across all models for the recall and F1 scores. Analysis showed that key improvement in recall was due to flipping of the *neutral* samples to other under-



(a)

(b)

(c)

Figure 2: (a) Recall vs. ℓ_t on goEmotions; increasing the extrapolative strength improves recall. (b) F1 vs. ℓ_t on tweetEmotions exhibits a similar trend (c) F1 vs. k for tweetEmotion using DeBERTa-xl; including layers 40 to 42 in the valid layers is found to be particularly useful.

Model	**Type**	**EmpatheticDialogue**			**tweetHate**			**tweetEmotion**			**goEmotions**			**Avg. F1**
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	
Flan-T5-large	✗	.551	.556	.543	.565	.577	.570	.298	.296	.286	.521	.469	.478	.469
Flan-T5-large	✓	.551	.557	.543	.579	.606	.590	.300	.299	.291	.513	.485	.487	.478
Flan-T5-large	β	.551	.557	.543	.590	.636	.610	.349	.311	.309	.493	.502	.489	.488
Flan-T5-xl	✗	.582	.569	.565	.566	.566	.559	.320	.300	.302	.503	.456	.465	.473
Flan-T5-xl	✓	.581	.570	.565	.690	.603	.615	.318	.314	.313	.499	.494	.486	.495
Flan-T5-xl	β	.582	.570	.565	.695	.605	.619	.316	.314	.313	.513	.494	.490	.497
DeBERTa-large	✗	.614	.601	.592	.647	.601	.622	.322	.299	.301	**.570**	.521	.534	.512
DeBERTa-large	✓	.616	.606	.597	.676	.643	.658	.313	.311	.308	.562	.536	.540	.526
DeBERTa-large	β	**.618**	.609	.601	.708	.675	**.690**	.312	.331	.319	.558	.543	**.541**	**.538**
DeBERTa-xl	✗	.604	.605	.590	.607	.596	.599	.324	.300	.303	.529	.493	.502	.498
DeBERTa-xl	✓	.610	.606	.594	**.727**	.668	.686	**.335**	.324	.325	.509	.530	.514	.523
DeBERTa-xl	β	.614	**.609**	**.597**	.725	**.668**	.685	.333	**.340**	**.334**	.505	**.555**	.522	.535

Table 1: Results of our experiments. ✗, ✓, and β each represent normal classification, static β, and dynamic β

represented classes. Appendix B shows the statistics of the flipped labels.

tweetEmotion: Contrastive classification with dynamic β performs significantly better over traditional classification. We see a general increase in recall and F1 with a slight harm to Precision. We also observed the emotions corrected by layer contrast were highly correlated. Appendix B contains more details about their statistics.

tweetHate: We see the maximum improvement in the performance of this dataset across all models. This improvement owes in large part to corrected predictions on underrepresented classes.

EmpatheticDialogue: For this dataset, we only see a slight increase in performance using the DeBERTa-xl model. Analyzing the probability distributions across layers, we observed no major change in probability distribution for different emotions across layers. The probability was distributed over a single label, increasing gradually across layers. This led to minimal contribution from layer contrast.

Effect of amateur layer selection: We use a bucket of layers for amateur layer selection defined by hyperparameter k. Figure 2c shows the trend of k against F1 using the DeBERTa-xl for tweetEmotions dataset. We observe that the performance generally increases up to a layer where the benefit of contrastive action is maximum, followed by a drop in performance. Upon evaluating early-exiting on intermediate layers, we observed that some layers are more adept at identifying specific classes than others, providing a variety of skills to contrast against for improved performance.

6 Conclusion

We propose a linear extrapolation approach for dynamically determining contrastive strength in layerwise contrastive decoding. Applied to fine-grained emotion classification tasks, this method enhances classifier performance by effectively addressing under-represented classes.This strengthens the promise of layer-contrast methods in domains other than text generation, and provides a technical contribution that reduces the variance of the method with respect to a core hyperparameter k, encouraging further research into how best to exploit the layerwise emergence of textual understanding to improve performance on a wide range of NLP tasks.

7 Limitations

Our study is restricted to fine-grained emotion classification with relatively small models (FLAN-T5 and DeBERTa). It remains to be seen whether our analysis of extrapolative classification will hold for prompt-based classification with larger models or across other datasets. We also found contrastive performance for smaller models to be sensitive to finetuning hyperparameters. Additionally, based on our results on EmpatheticDialogue we observe that CD tends to work better when model uncertainty is high i.e. probability distribution across labels changes more often across layers as shown in Figure 4. Extending the method to identify and better handle these cases is left to future work.

References

Dimosthenis Antypas, Asahi Ushio, Francesco Barbieri, Leonardo Neves, Kiamehr Rezaee, Luis Espinosa-Anke, Jiaxin Pei, and Jose Camacho-Collados. 2023. SuperTweetEval: A challenging, unified and heterogeneous benchmark for social media NLP research. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12590–12607, Singapore. Association for Computational Linguistics.

Francesco Barbieri, José Camacho-Collados, Leonardo Neves, and Luis Espinosa-Anke. 2020. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. *ArXiv*, abs/2010.12421.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. SemEval-2019 task 3: EmoContext contextual emotion detection in text. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 39–48, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *ArXiv*, abs/2309.03883.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics.

Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. 2020. Depth-adaptive transformer. In *International Conference on Learning Representations*.

Yao Fu, Litu Ou, Mingyu Chen, Yuhao Wan, Hao Peng, and Tushar Khot. 2023. Chain-of-thought hub: A continuous effort to measure large language models' reasoning performance.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. {DEBERTA}: {DECODING}-{enhanced} {bert} {with} {disentangled} {attention}. In *International Conference on Learning Representations*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq R. Joty, Richard Socher, and Nazneen Rajani. 2020. Gedi: Generative discriminator guided sequence generation. In *Conference on Empirical Methods in Natural Language Processing*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2022. Contrastive decoding: Open-ended text generation as optimization. In *Annual Meeting of the Association for Computational Linguistics*.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. Dexperts: Decoding-time controlled text generation with experts and anti-experts. In *Annual Meeting of the Association for Computational Linguistics*.

Nikolay Malkin, Zhen Wang, and Nebojsa Jojic. 2021. Coherence boosting: When your pretrained language model is not paying enough attention. In *Annual Meeting of the Association for Computational Linguistics*.

J. A. Meaney, Steven Wilson, Luis Chiruzzo, Adam Lopez, and Walid Magdy. 2021. SemEval 2021 task 7: HaHackathon, detecting and rating humor and offense. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 105–119, Online. Association for Computational Linguistics.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 task 1: Affect in tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 1–17, New Orleans, Louisiana. Association for Computational Linguistics.

Sean O'Brien and Mike Lewis. 2023. Contrastive decoding improves reasoning in large language models. *ArXiv*, abs/2309.09117.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada. Association for Computational Linguistics.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. 2022. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*.

Mayukh Sharma, Ilanthenral Kandasamy, and W.B. Vasantha. 2023. Emotion quantification and classification using the neutrosophic approach to deep learning. *Applied Soft Computing*, 148:110896.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Scott Yih. 2023. Trusting your evidence: Hallucinate less with context-aware decoding. *ArXiv*, abs/2305.14739.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Surat Teerapittayanon, Bradley McDanel, and H.T. Kung. 2016. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2464–2469.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, Yidong Wang, Linyi Yang, Jindong Wang, Xing Xie, Zheng Zhang, and Yue Zhang. 2023. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *ArXiv*, abs/2310.07521.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed Huai hsin Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

G. Yona, Or Honovich, Itay Laish, and Roee Aharoni. 2023. Surfacing biases in large language models using contrastive input decoding. *ArXiv*, abs/2305.07378.

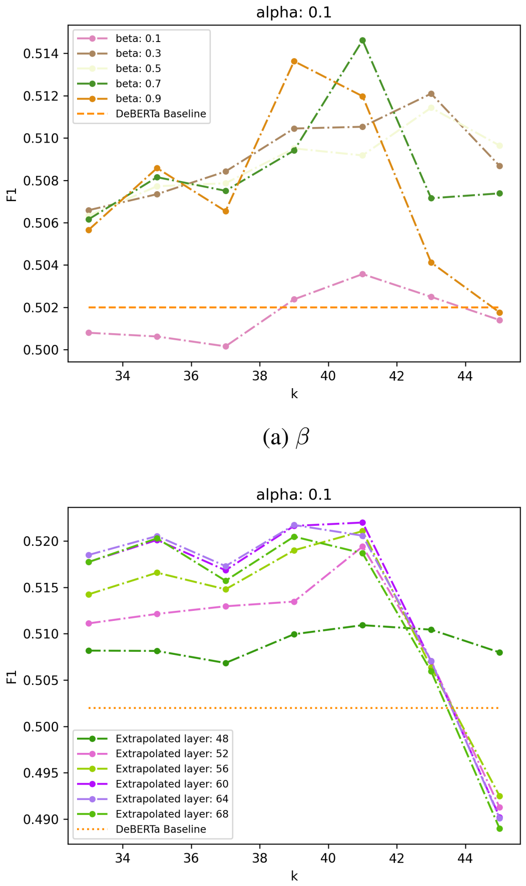
Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing. 2023. Sentiment analysis in the era of large language models: A reality check.

A Hyperparameter Sweep for k

Table 2 contains the values of hyperparameter k used for reporting the results. We also show the effect of k on performance for the goEmotions dataset using both β and dynamic β in Figure 3.

Model	goEmotions	tweetEmotion	tweetHate	Empathetic Dialogue
Flan-T5-large	19	20	17	15
Flan-T5-xl	15	15	17	15
DeBERTa-large	15	19	17	19
DeBERTa-xl	39	41	38	43

Table 2: Our choice of hyperparameter k for defining the amateur search space used in the final results. The final layer is 48 for DeBERTa-xl and 23 for the remaining models.



(a) β

(b) Dynamic β

Figure 3: Effect of k against β and dynamic β for goEmotions dataset using DeBERTa-xlarge. The trend is more stable with dynamic β.

B Analysis of corrected samples

Table 3 shows the frequency of correctly flipped samples (true positives) vs. correctly flipped samples (positives) from the neutral class (wrongly predicted as neutral). We observe that neutral forms the majority of samples flipped to other underrepresented classes. Table 4 contains the count of emotions that were correctly flipped from neutral.

Model	Total	Neutral
Flan-T5-large	105	80/105
Flan-T5-xl	72	57/72
DeBERTa-large	74	51/74
DeBERTa-xl	164	130/164

Table 3: Count of correctly flipped samples (all emotions classes) vs. correctly flipped samples only from the neutral class.

From	To	Count
neutral	disapproval	22
neutral	curiosity	19
neutral	annoyance	13
neutral	admiration	12
neutral	approval	11

Table 4: Count of samples moved from neutral to other classes for goEMotions using DeBERTa-xl.

We also report the most frequent samples corrected for the tweetEmotion dataset using layer contrast (dynamic β). We see that the emotions for the pair of corrected samples were highly correlated.

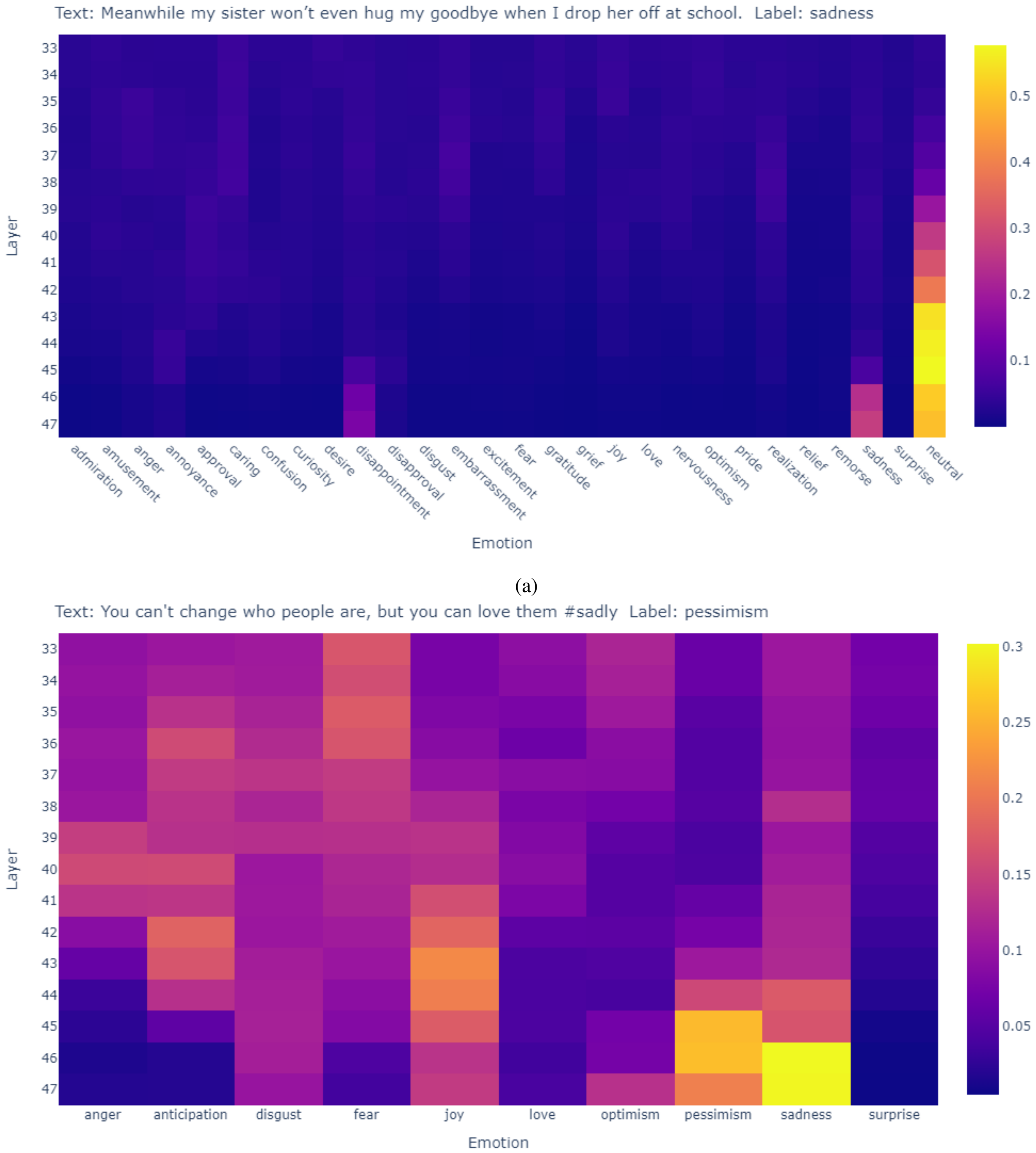
Model	Emotion
Flan-T5-large	sadness $\mapsto$ pessimism: 7 joy $\mapsto$ anticipation: 6
Flan-T5-xl	sadness $\mapsto$ pessimism: 19 anger $\mapsto$ disgust: 17
deBERTa-large	anger $\mapsto$ disgust: 15 joy $\mapsto$ anticipation: 8
deBERTa-xl	sadness $\mapsto$ pessimism: 20 joy $\mapsto$ optimism: 7

Table 5: Count of top 2 emotion pairs that were contrastively flipped for each model.

C Knowledge pattern across layers

Fine-grained emotion analysis is challenging due to non-mutually exclusive labels and similar polarity among different emotions, making it hard to accurately classify them. Class imbalance further biases the model towards more frequent emotions.

To study the change in probability distribution for emotions across layers, we performed early exiting on different layers of our fine-tuned models



(a)

(b)

Figure 4: Probability distribution across the finetuned layers of DeBERTa-xl for a sample from each (a) goEmotions and (b) tweetEmotion dataset. In the goEmotions sample, the model initially identifies the label as neutral but increases the probabilities assigned to sadness and disappointment (the true label) over subsequent layers. For the tweetEmotion sample, the probability distribution changes across layers and the model fails to assign a high probability to a single emotion.

to visualize how the distributions across emotions evolve. We observed that for some emotions, the model makes a decision very early, passing it along the layers without much change. For others, the distribution tends to change in later layers, suggesting that the model is still adding information. We observed this pattern mostly around classes that are rarer in the training data or more closely related to each other. Figure 4 shows the change in distribution for two examples.

Drawing from these observations, we combine the idea of contrastive decoding and DoLa for fine-grained emotion analysis. We build on DoLa, using the early exited intermediate layers as amateur models. We then use contrastive action against the final layer distribution chosen as our expert model. Additionally, we deduce a method to dynamically select the contrastive strength which we show leads to better performance on fine-grained emotion tasks.

D Dataset Details

goEmotions (Demszky et al., 2020) introduces a new emotion taxonomy of emotions named goEmotions consisting of 28 emotions with neutral. The 27 emotion classes are fine-grained over 7 emotions defined in Ekman taxonomy. It contains roughly 58k samples overcoming the problems with earlier emotion datasets which were small in size and covered a very limited taxonomy. The dataset contained a few multilabel data-points, which we filter out for our experiments.

SuperTweetEval (Antypas et al., 2023) aims to provide a unified benchmark to evaluate the performance of models on NLP tasks across social media. It is a heterogeneous collection of multiple datasets spanning NER, QA, and classification. For our experiments, we use tweetEmotion and tweetHate focused on multi-class classification, with each dataset containing 12 and 8 classes.

EmpatheticDialogues (Rashkin et al., 2019) was introduced as a benchmark for training and evaluating models and their capability to understand and acknowledge empathetic text. The dataset contains conversations distributed across 32 emotions. We use the first text of the conversation and the corresponding emotion for defining our fine-grained classification task.

E Computational Resources Estimate

Early compute was run on freely available Cloud T4 GPUs. Fine-tuning and later experiments were run on a cluster of A6000 GPUs, with a maximum of 8 used at a single time.

Fine-tuning all models across all datasets takes roughly 2 GPU-hours. Hyperparameter searches are performed at classification time, which takes very little compute. A very rough estimate for GPU-hours in this project is 50.