Understanding the Robustness of Multi-modal Contrastive Learning to Distribution Shift

Anonymous Authors

Reviewed on OpenReview: NA

Abstract

Recently, multimodal contrastive learning (MMCL) approaches, such as CLIP (Radford et al., 2021), have achieved a remarkable success in learning representations that are robust against distribution shift and generalize to new domains. Despite the empirical success, the mechanism behind learning such generalizable representations is not understood. In this work, we rigorously analyze this problem and uncover two mechanisms behind MMCL's robustness: *intra-class contrasting*, which allows the model to learn features with a high variance, and *inter-class feature sharing*, where annotated details in one class help learning other classes better. Both mechanisms prevent spurious features that are over-represented in the training data to overshadow the generalizable core features. This yields superior zero-shot classification accuracy under distribution shift. Furthermore, we theoretically demonstrate the benefits of using rich captions on robustness and explore the effect of annotating different types of details in the captions. We validate our theoretical findings through experiments, including a well-designed synthetic experiment and an experiment involving training CLIP on MS COCO (Lin et al., 2014) and evaluating the model on variations of shifted ImageNet.

Keywords: Multi-modal learning, Contrastive learning, Distribution shift, Robustness

1 Introduction

The pursuit of learning representations that generalize well under distribution shifts and across different domains remains a significant challenge in machine learning. Traditional Supervised Learning (SL) tends to capture simpler, domain-specific features rather than complex, generalizable ones. In contrast, recent advancements in Multimodal Contrastive Learning (MMCL), particularly with large-scale image-text pair datasets, have shown exceptional zero-shot classification abilities and domain transferability. MMCL's success, showcased by models like CLIP Radford et al. (2021) and ALIGN Jia et al. (2021), comes from a training process that aligns the representations of each paired image and text while also contrasting those of non-paired ones. Despite its effectiveness, the reasons behind MMCL's robustness over SL are still not fully comprehended.

This study aims to unravel the mechanisms behind MMCL's robust feature learning and to understand how they contribute to robust zero-shot classification. We present evidence that its robustness primarily stems from the MMCL loss function and the textual supervision of images. We identify two key mechanisms: intra-class contrasting, which encourages the learning of variable yet generalizable features within the same latent class, and inter-class feature sharing, which enables the learning of information about a latent class through its occurrence in other classes. These mechanisms collectively empower MMCL to learn representations that, when leveraged by zero-shot classification, enhance robustness against distribution shifts. Our experiments, including one with synthetic data, and others where we

train models on MS COCO Lin et al. (2014) and Conceptual Captions Sharma et al. (2018) and then evaluate them on six shifted versions of ImageNet, corroborate the pivotal role of MMCL loss and rich text annotations in achieving robustness. These findings reinforce our theoretical insights.

2 Theoretical Analysis

A framework for comparing MMCL and SL. We present a general framework for comparing MMCL and SL in Section B, along with the corresponding notations. We start by modeling the multimodal data, and then formalize the MMCL and SL pipelines and their evaluation for robustness to distribution shift.

Two mechanisms behind the robustness of MMCL. Next, we formulate and analyze specific types of distribution shift. In Section C we explore two scenarios illustrating MMCL's superior robustness to distribution shift, compared to SL. (Thm5&6) We start by analyzing the first scenario which illustrates how MMCL can learn features that are challenging for SL to learn. Consider the case where the majority or all images of a 'cow' appear on 'grass'. Here, grass is a spurious feature with high correlation with cow. Grass is often a simple green surface without a high variation. But, cows can vary a lot in their appearance. This makes cows more difficult to learn than grass. Sagawa et al. (2020) have demonstrated that SL tends to learn the simple spurious feature (grass), whereas our findings, as outlined in Thm 6, reveal that MMCL can successfully learn the core feature (cow) with large variance. This is because the loss function, in conjunction with detailed captions, inherently encourages contrasting between individual cows, thereby facilitating the learning of specifics for each cow. (Thm9&10) Next, we consider the second scenario and demonstrate how MMCL benefits from annotated details in some latent classes to disassociate spurious correlations in other latent classes, while SL fails to grasp these details. For example, typical images of a 'tree' have green leaves. However, trees in in the background of images of 'wolf' or 'ski resort' may appear without leaves. SL, which only observes the labels, tends to overlook the trees without leaves as they do not contribute to learning 'wolf' and 'ski resort', thus incorrectly correlating trees with the color green. In contrast, in MMCL, if the trees without leaves are annotated in the captions, the model can disassociate the green leaves from tree.

Understand the benefit of rich image captions. We delve into the role of richness of captions in robustness. Two conclusions. In Thm 12 we show that mentioning variations of the core feature benefits robustnres through intra-class contrasting. In Thm 14 we show that mentioning more features in general benefits robustness through inter-class feature sharing.

3 Experiments

We conduct experiments on carefully designed semi-synthetic data, MS COCO and Conceptual Captions. We find that (1) reducing the richness of captions by replacing them with labels worsens robustness, and (2) decreasing the amount of inter-class contrasting by removing negative pairs from different classes from the loss also leads to decreased robustness. These findings validate our theoretical insights in Theorems 6, 5, 12 and 14.

References

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.

Benjamin Aubin, Agnieszka Słowik, Martin Arjovsky, Leon Bottou, and David Lopez-Paz. Linear unit-tests for invariance discovery. *arXiv preprint arXiv:2102.10867*, 2021.

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in neural information processing systems*, 32, 2019.

Tianle Cai, Ruiqi Gao, Jason Lee, and Qi Lei. A theory of label propagation for subpopulation shift. In *International Conference on Machine Learning*, pages 1170–1182. PMLR, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. February 2020. doi: 10.48550/arXiv.2002.05709. URL `https://arxiv.org/abs/2002.05709v3`.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased Contrastive Learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 8765–8775. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/63c3ddcc7b23daa1e42dc41f9a44a873-Abstract.html`.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Matthew Fahrbach, Adel Javanmard, Vahab Mirrokni, and Pratik Worah. Learning rate schedules in the presence of distribution shift. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 9523–9546. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/fahrbach23a.html`.

Alex Fang, Gabriel Ilharco, Mitchell Wortsman, Yuhao Wan, Vaishaal Shankar, Achal Dave, and Ludwig Schmidt. Data determines distributional robustness in contrastive language image pre-training (clip). In *International Conference on Machine Learning*, pages 6216–6234. PMLR, 2022.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan Rossi, Vishwa Vinay, and Aditya Grover. Cyclip: Cyclic contrastive language-image pretraining. *Advances in Neural Information Processing Systems*, 35:6704–6719, 2022.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34:5000–5011, 2021.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021b.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

Wenlong Ji, Zhun Deng, Ryumei Nakada, James Zou, and Linjun Zhang. The power of contrast for feature learning: A theoretical analysis. *arXiv preprint arXiv:2110.02473*, 2021.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021.

Siddharth Joshi, Yu Yang, Yihao Xue, Wenhan Yang, and Baharan Mirzasoleiman. Towards mitigating spurious correlations in the wild: A benchmark and a more realistic dataset, 2023.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pages 121–137. Springer, 2020.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

Stephen J Montgomery-Smith. The distribution of rademacher sums. *Proceedings of the American Mathematical Society*, 109(2):517–522, 1990.

Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *European Conference on Computer Vision*, pages 529–544. Springer, 2022.

Ryumei Nakada, Halil Ibrahim Gulluk, Zhun Deng, Wenlong Ji, James Zou, and Linjun Zhang. Understanding multimodal contrastive learning and incorporating unpaired data. *arXiv preprint arXiv:2302.06232*, 2023.

Edwin G Ng, Bo Pang, Piyush Sharma, and Radu Soricut. Understanding guided image captioning performance across domains. *arXiv preprint arXiv:2012.02339*, 2020.

Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.

Thao Nguyen, Samir Yitzhak Gadre, Gabriel Ilharco, Sewoong Oh, and Ludwig Schmidt. Improving multimodal datasets with image captioning. *arXiv preprint arXiv:2307.10350*, 2023.

Hieu Pham, Zihang Dai, Golnaz Ghiasi, Kenji Kawaguchi, Hanxiao Liu, Adams Wei Yu, Jiahui Yu, Yi-Ting Chen, Minh-Thang Luong, Yonghui Wu, et al. Combined scaling for zero-shot transfer learning. *Neurocomputing*, 555:126658, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019.

Yunwei Ren and Yuanzhi Li. On the importance of contrastive loss in multimodal learning. *arXiv preprint arXiv:2304.03717*, 2023.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015a. doi: 10.1007/s11263-015-0816-y.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015b.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pages 8346–8356. PMLR, 2020.

Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. Breeds: Benchmarks for subpopulation shift. *arXiv preprint arXiv:2008.04859*, 2020.

Shibani Santurkar, Yann Dubois, Rohan Taori, Percy Liang, and Tatsunori Hashimoto. Is a caption worth a thousand images? a controlled study for representation learning. *arXiv preprint arXiv:2207.07635*, 2022.

Nikunj Saunshi, Jordan Ash, Surbhi Goel, Dipendra Misra, Cyril Zhang, Sanjeev Arora, Sham Kakade, and Akshay Krishnamurthy. Understanding contrastive learning requires incorporating inductive biases. *arXiv preprint arXiv:2202.14037*, 2022.

Vaishaal Shankar, Achal Dave, Rebecca Roelofs, Deva Ramanan, Benjamin Recht, and Ludwig Schmidt. Do image classifiers generalize across time? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9661–9669, 2021.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.

Galen R Shorack and GR Shorack. *Probability for statisticians*, volume 951. Springer, 2000.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

Nitish Srivastava and Russ R Salakhutdinov. Multimodal learning with deep boltzmann machines. *Advances in neural information processing systems*, 25, 2012.

Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599, 2020.

Yuandong Tian. Understanding deep contrastive learning via coordinate-wise optimization. *Advances in Neural Information Processing Systems*, 35:19511–19522, 2022.

Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, pages 10268–10278. PMLR, 2021.

Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive estimation reveals topic posterior information to linear models. *J. Mach. Learn. Res.*, 22:281–1, 2021a.

Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive learning, multi-view redundancy, and linear models. In *Algorithmic Learning Theory*, pages 1179–1206. PMLR, 2021b.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

Yoav Wald, Amir Feder, Daniel Greenfeld, and Uri Shalit. On calibration and out-of-domain generalization. *Advances in neural information processing systems*, 34:2215–2227, 2021.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pages 10506–10518, 2019.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

Per-Åke Wedin. Perturbation bounds in connection with singular value decomposition. *BIT Numerical Mathematics*, 12:99–111, 1972.

Zixin Wen and Yuanzhi Li. Toward understanding the feature learning process of self-supervised contrastive learning. In *International Conference on Machine Learning*, pages 11112–11122. PMLR, 2021.

Ruijia Wu, Linjun Zhang, and T Tony Cai. Sparse topic modeling: Computational efficiency, near-optimal algorithms, and statistical inference. *Journal of the American Statistical Association*, pages 1–13, 2022.

Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint arXiv:2006.09994*, 2020.

Yihao Xue, Siddharth Joshi, Eric Gan, Pin-Yu Chen, and Baharan Mirzasoleiman. Which features are learnt by contrastive learning? on the role of simplicity bias in class collapse and feature suppression. *arXiv preprint arXiv:2305.16536*, 2023.

Yuzhe Yang, Haoran Zhang, Dina Katabi, and Marzyeh Ghassemi. Change is hard: A closer look at subpopulation shift. *arXiv preprint arXiv:2302.12254*, 2023.

Huaxiu Yao, Linjun Zhang, and Chelsea Finn. Meta-learning with fewer tasks through task interpolation. *arXiv preprint arXiv:2106.02695*, 2021.

Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving out-of-distribution robustness via selective augmentation. In *International Conference on Machine Learning*, pages 25407–25437. PMLR, 2022.

Xingxuan Zhang, Yue He, Renzhe Xu, Han Yu, Zheyan Shen, and Peng Cui. Nico++: Towards better benchmarking for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16036–16047, 2023.

Zhuotun Zhu, Lingxi Xie, and Alan L Yuille. Object recognition with and without objects. *arXiv preprint arXiv:1611.06596*, 2016.

Appendix A. Related Works

Distribution shift. There is a long line of work on dealing with different types of distribution shift. This includes sub-population shift and domain generalization among others, where distribution of sub-populations in training and test data is different, and some sub-populations may be underrepresented or missing in the training data (Cai et al., 2021; Yang et al., 2023; Santurkar et al., 2020), (Gulrajani and Lopez-Paz, 2020; Joshi et al., 2023; Zhang et al., 2023; Hu et al., 2020; Fahrbach et al., 2023), or a hybrid of both Koh et al. (2021). Another line of research focuses on evaluating models on natural variations in the source of data collection, with the precise category of shift typically unformalized or unknown. For example, a dataset that contains art and cartoon renditions of ImageNet classes (Hendrycks et al., 2021a), and other variations of ImageNet (Barbu et al., 2019; Recht et al., 2019; Shankar et al., 2021). Despite the diversity of settings, extensive studies (Sagawa et al., 2019, 2020; Xiao et al., 2020; Ilyas et al., 2019) revealed a common theme across these subfields: deep learning models often rely heavily on *spurious correlations* that are specific to the training data but do not hold in general, e.g., those between certain object classes and backgrounds/textures in the image (Zhu et al., 2016; Geirhos et al., 2018).

Multi-modal learning. Learning better representations based on multiple modalities has been a long pursuit (Ngiam et al., 2011; Srivastava and Salakhutdinov, 2012). Numerous methods for learning joint vision-language representations (Li et al., 2019; Lu et al., 2019; Tan and Bansal, 2019; Li et al., 2020; Yao et al., 2021) have emerged. Among them, MMCL (Radford et al., 2021; Jia et al., 2021; Mu et al., 2022; Goel et al., 2022; Pham et al., 2023) has stood out by achieving SOTA performance in various tasks. Notably, Radford et al. (2021) showed that MMCL on large image-text datasets achieves a significant improvement in robustness to distribution shift. The empirical investigations of Fang et al. (2022) suggests that this is only attributed to the large diverse image training data, with MMCL loss and text supervision contributing little. We show *provably* that it is not *only* the diverse image data that contributes to superior robustness of MMCL. Indeed, MMCL loss and richness of text annotations are crucial factors.

(Uni-modal) CL. There have been both empirical Chen et al. (2020); Chuang et al. (2020); Khosla et al. (2020) and theoretical studies about CL Wang and Isola (2020); Tosh et al. (2021a,b); Arora et al. (2019); HaoChen et al. (2021); Wen and Li (2021); Ji et al. (2021); Saunshi et al. (2022); Xue et al. (2023). We will demonstrate for the first time how the contrastive aspect of the loss can benefit robustness against distribution shift, while noting that this advantage only exists when equipped with multi-modality and zero-shot learning, which are present in MMCL but not in unimodal CL.

Appendix B. A Framework for Comparing MMCL and SL

In this section, we present a general framework for comparing MMCL and SL, along with the corresponding notations. We start by modeling the multimodal data, and then formalize the MMCL and SL pipelines and their evaluation for robustness to distribution shift. We will formulate and analyze specific types of distribution shift in the next section.

B.1 Modeling Multimodal Data

To model multimodal data, it is essential to capture the fact that inputs from different modalities can represent the same abstract notion. For instance, both text and an image

can represent 'a cow on grass'. We define *underlying feature vectors* to model this abstract notion, and model each input in a specific modality as a projection of the underlying feature vector onto that modality's input space.

Underlying feature . There is an underlying feature space shared among different modalities, where abstract notions reside. We model this as a vector space $\mathbb{R}^l$, where each vector is termed an *underlying feature vector* (e.g., 'a cow on grass'), and each element within the vector is referred to as an *underlying feature* (e.g., 'cow').

Latent classes and labels. Each $\boldsymbol{z}$ is associated with a *latent class*, represented by a *label* y. We note that the labels are only used by SL but not by MMCL.

Inputs in each modality. Each input example in a modality is an instantiation of an abstract notion. We model this as a projection from an underlying feature vector to another space where this modality's inputs live. Formally, let M represent a modality. Given a underlying feature vector $\boldsymbol{z}$, a corresponding input $\boldsymbol{x}_M$ in this modality is generated as: $\boldsymbol{x}_M = \boldsymbol{D}_M \boldsymbol{\mu}_M(\boldsymbol{z}) + \boldsymbol{\xi}_M$, where $\boldsymbol{\mu}_M(\boldsymbol{z}) \in \mathbb{R}^l$ is a random vector that depends on $\boldsymbol{z}$. It can be interpreted as a possibly distorted version of the original feature vector $\boldsymbol{z}$. Note that setting $\boldsymbol{\mu}_M(\boldsymbol{z}) = \boldsymbol{z}$ implies no distortion in the features when represented in this modality. $\boldsymbol{\xi}_M \in \mathbb{R}^{d_M}$ is a random noise drawn from $\mathcal{N}(0, \frac{\sigma_\xi^2}{d_M}\mathbf{I}_{d_M})$. The matrix $\boldsymbol{D}_M \in \mathbb{R}^{d_M \times l}$ $(d_M > l)$ is a matrix with orthonormal columns that can be interpreted as a dictionary matrix. It captures the transformation from the lower dimensional feature space to the higher dimensional input space. Different modalities can have different $\boldsymbol{D}_M$ matrices, reflecting the idea that the same underlying feature may be instantiated differently in each modality (e.g., colors are represented differently in images and texts). Modeling modalities as above is consistent with (Nakada et al., 2023).

In this paper, for clarity and illustration, we focus on the popular vision and language modalities. We let I denote the modality for images, and T denote the modality for texts. However, we note that our framework and results directly apply to other modalities.

Distribution shift. We define two joint distributions between underlying features and latent classes: $\mathscr{P}^\star$, representing the 'ground-truth' in the real world, and $\mathscr{P}^{Tr}$, from which our training data are drawn. We let $\mathscr{P}^{Tr}$ exhibit spurious correlations between certain features and latent classes which do not hold in the ground-truth $\mathscr{P}^\star$. This setup captures the underlying reason for the performance drop observed in various types of distribution shift scenarios, as we discuss in Section A.

B.2 Multi-Modal Contrastive Learning (MMCL)

Unlike traditional supervised learning, MMCL does not see the input data's latent classes, but is instead given pairs of inputs from two modalities and aims to learn the correspondence between them.

Training dataset. The training dataset comprises n image-text pairs, denoted as $\{(\boldsymbol{x}_{I,i}, \boldsymbol{x}_{T,i})\}_{i=1}^n$, where for each index i, both $\boldsymbol{x}_{I,i}$ and $\boldsymbol{x}_{T,i}$ are generated based on the same underlying feature vector $\boldsymbol{z}_i$. In practice, the texts are usually captions accompanying the images. The feature vectors $\{\boldsymbol{z}_i\}_{i=1}^n$ are drawn from the training distribution $\mathscr{C}^{Tr}$.

Linear encoders. The encoders for modalities I and T are denoted as $g_I : \mathbb{R}^{d_I} \to \mathbb{R}^p$ and $g_T : \mathbb{R}^{d_T} \to \mathbb{R}^p$ respectively. We consider linear models for the encoders, given by $g_I(\boldsymbol{x}) = \boldsymbol{W}_I \boldsymbol{x}$ and $g_T(\boldsymbol{x}) = \boldsymbol{W}_T \boldsymbol{x}$, where $\boldsymbol{W}_I \in \mathbb{R}^{d_I \times p}$ and $\boldsymbol{W}_T \in \mathbb{R}^{d_T \times p}$ with $p \geq l$ are the corresponding encoder parameters. Linear encoders are employed widely in previous studies

of MMCL (Nakada et al., 2023; Ren and Li, 2023) and general feature learning (Jing et al., 2021; Tian et al., 2021; Ji et al., 2021; Wu et al., 2022; Tian, 2022; Xue et al., 2023), to facilitate the analysis. In Section E, we will empirically confirm that our findings extend to non-linear models.

Representation learning with MMCL. MMCL learns representations for both modalities in a shared latent space. We consider the linearized contrastive loss function from Nakada et al. (2023):

$$\mathcal{L}_{\text{MMCL}}(\boldsymbol{W}_I,\boldsymbol{W}_T)=\frac{1}{2n(n-1)}\sum_i\sum_{j\neq i}(s_{ij}-s_{ii})+\frac{1}{2n(n-1)}\sum_i\sum_{j\neq i}(s_{ji}-s_{ii})+\frac{\rho}{2}\|\boldsymbol{W}_I^\top\boldsymbol{W}_T\|_F^2,$$

where $s_{ij} \coloneqq g_I(\boldsymbol{x}_{I,i})^\top g_T(\boldsymbol{x}_{T,j}) = (\boldsymbol{W}_I\boldsymbol{x}_{I,i})^\top\boldsymbol{W}_T\boldsymbol{x}_{T,j}$ is the similarity (measured by inner product) between representations of an image and a text. This loss encourages the model to *align* each image-text pair by increasing their representation similarity (s_{ii}) and *contrast* between images and texts that are not paired together by reducing their representation similarity ($s_{ij}, i \neq j$). The last term is a regularization term with $\rho > 0$. The linear loss and its uni-modal counterpart are widely used in analysis of CL, as they closely captures the dynamics of popular contrastive losses (Ji et al., 2021; Tian, 2022; Nakada et al., 2023), such as CLIP, as we will experimentally confirm in Section E.

Prompts for zero-shot classification. We test the model's capability in zero-shot classification, where a text prompt is created for each label (e.g., 'a photo of a *dog*'), and the prediction is determined by the prompt with the highest representation similarity with the given image. To formalize this, we define the prompt $\boldsymbol{p}_y$ for each latent class y as $\boldsymbol{p}_y = \boldsymbol{D}_T\bar{\boldsymbol{z}}_y$, where $\bar{\boldsymbol{z}}_y \coloneqq \mathbb{E}_{(\boldsymbol{z},y)\sim\mathscr{P}^\star}[\boldsymbol{z}|y]$. That is, the prompt is 'the center of all underlying feature vectors with label y in the true distribution' represented in modality T. This closely resembles real world practices where the representation of multiple texts with engineered templates like `'a bad photo of a {}', 'a good photo of a {}'` are averaged (Radford et al., 2021).

Robustness evaluation. Given two encoders g_I and g_T with parameters $\boldsymbol{W}_I$ and $\boldsymbol{W}_T$, respectively, we evaluate the zero-shot performance on the true distribution $\mathscr{P}^\star$. Formally, given an image $\boldsymbol{x}_I$, the prediction is $\hat{y}(\boldsymbol{x}_I) = \arg\max_y g_I(\boldsymbol{x}_I)^\top g_T(\boldsymbol{p}_y)$. The test accuracy, denoted by $\text{Acc}^{\text{MMCL}}_{\mathscr{P}^\star}(\boldsymbol{W}_I,\boldsymbol{W}_T)$, is computed as

$$\text{Acc}^{\text{MMCL}}_{\mathscr{P}^\star}(\boldsymbol{W}_I,\boldsymbol{W}_T) = \mathbb{E}_{(\boldsymbol{z},y)\in\mathscr{P}^\star,\boldsymbol{x}_I=\boldsymbol{D}_I\boldsymbol{\mu}(\boldsymbol{z})+\boldsymbol{\xi}_I}[\mathbb{1}(\hat{y}(\boldsymbol{x}_I)=y)], \tag{1}$$

where $\mathbb{1}(\cdot)$ denotes the indicator function.

Relation to feature cross-covariance. We utilize the connection between the cross-covariance between images and captions, and the MMCL objective for our analysis.

Definition 1 *We define $\boldsymbol{C}^{Tr}$ as the cross-covariance between images' and texts' feature vectors $\boldsymbol{C}^{Tr} \coloneqq \mathbb{E}_{\boldsymbol{z}\in\mathscr{C}^{Tr},\,\boldsymbol{\mu}_I(\boldsymbol{z}),\,\boldsymbol{\mu}_T(\boldsymbol{z})}[\boldsymbol{\mu}_I(\boldsymbol{z})\boldsymbol{\mu}_T(\boldsymbol{z})^\top]$. When $\boldsymbol{\mu}_I(\cdot)$ and $\boldsymbol{\mu}_T(\cdot)$ both are identity, $\boldsymbol{C}^{Tr}$ is the covariance of the original feature vector.*

Lemma 2 (Informal) *Given an image with feature $\boldsymbol{\mu}'$ and a text with feature $\boldsymbol{\mu}''$, the similarity (inner product of representations) between them, computed using encoders trained on the training set, is: **similarity score** $\approx \boldsymbol{\mu}'^\top\boldsymbol{C}^{Tr}\boldsymbol{\mu}'' = \sum_{i=1}^l\sum_{j=1}^l C^{Tr}_{ij}\mu'_i\mu''_j$.*

That is, the image-text similarity is a weighted sum of products between the features in image and text inputs. The weights are determined by the feature cross-covariance matrix of training data, whose i, j-th element is the covariance between feature i in images and feature j in texts.

Importance of zero-shot. We emphasize that using zero-shot classification instead of training a linear classifier on the representations is crucial for achieving robustness in MMCL. The latter essentially involves SL, which falls short for the same reasons as shown in our analysis for SL in Section C.

B.3 Supervised Learning (SL)

Standard SL has access to each input's label and the inputs are from a single modality (i.e., images). Let $\{(\boldsymbol{x}_{I,i}, y_i)\}_{i=1}^n$ be the training dataset with n inputs $\boldsymbol{x}_{I,i}$ and their labels y_i, we train a linear model $f(\boldsymbol{x}) = \boldsymbol{W}^\top \boldsymbol{x}$ with weights $\boldsymbol{W} \in \mathbb{R}^{d_I \times q}$, where $q = 1$ for binary classification and $q = \#$classes for multi-class classification. We consider minimizing logistic loss for binary classification, and Cross-Entropy loss for multiclass classification, with gradient descent at a sufficiently small step size.

Robustness evaluation. For testing, given a model with weights $\boldsymbol{W}$, the accuracy, denoted by $\mathrm{Acc}^{\mathrm{SL}}_{\mathscr{P}^\star}(\boldsymbol{W})$, is evaluated on the true distribiution $\mathscr{P}^\star$ as $\mathrm{Acc}^{\mathrm{SL}}_{\mathscr{P}^\star}(\boldsymbol{W}) = \mathbb{E}_{(\boldsymbol{z},y)\in\mathscr{P}^\star, \boldsymbol{x}_I = \boldsymbol{D}_I\boldsymbol{\mu}(\boldsymbol{z})+\boldsymbol{\xi}_I}[\mathbb{1}(\hat{y}(\boldsymbol{x}_I) = y)]$, where $\hat{y}(\boldsymbol{x}_I) = \mathrm{sign}(\boldsymbol{W}^\top \boldsymbol{x}_{I_j})$ for binary classification, and $\hat{y}(\boldsymbol{x}_I) = \arg\max_j [\boldsymbol{W}^\top \boldsymbol{x}_I]_j$ for multi-class classification, with $[\boldsymbol{W}^\top \boldsymbol{x}_I]_j$ denoting the j-th element in the vector $\boldsymbol{W}^\top \boldsymbol{x}_I$.

Appendix C. Two Mechanisms behind the Robustness of MMCL

Next, we explore two scenarios illustrating MMCL's superior robustness to distribution shift, compared to SL. First, we consider the scenario where generalizable core feature has a higher variance than domain-dependent spurious feature. Then, we consider the data distribution where each latent class has a core feature, that co-occurs with a strong spurious feature in the training data. These features can occur in other latent classes as well, independently of each other. For clarity, we set $\boldsymbol{\mu}_I(\boldsymbol{z}) = \boldsymbol{z}, \boldsymbol{\mu}_T(\boldsymbol{z}) = \boldsymbol{z}, \forall \boldsymbol{z} \in \mathbb{R}^l$ in this section.

C.1 Robustness via Intra-class Contrasting

We start by analyzing the first scenario which illustrates how MMCL can learn features that are challenging for SL to learn. Consider the case where the majority or all images of a 'cow' appear on 'grass'. Here, grass is a spurious feature with high correlation with cow. Grass is often a simple green surface without a high variation. But, cows can vary a lot in their appearance. This makes cows more difficult to learn than grass. Below, we will formalize this scenario and demonstrate that SL learns the spurious feature (grass) but MMCL learns the generalizable feature (cow) and obtains a superior robustness.

C.1.1 Distribution of features

The following definition simulates the aforementioned scenario.

Definition 3 (Data Model 1) *In both $\mathscr{P}^\star$ and $\mathscr{P}^{Tr}$, each label y is uniformly drawn from $\{-1, 1\}$ and the corresponding feature vector $\boldsymbol{z} \in \mathbb{R}^2$ is generated as $\boldsymbol{z} = [z_{core}, z_{spu}]^T$ where $z_{core} \sim \mathcal{N}(y, \sigma^2_{core})$, represents the core feature that contains information of the label y, and*

$z_{spu} \sim \mathcal{N}(a, \sigma_{spu}^2)$. *In the true distribution* $\mathscr{P}^\star$, *a is uniformly drawn from* $\{-1, 1\}$ *and is independent of the label* y, *making the feature* z_{spu} *irrelevant to the label. However, in the training distribution* $\mathscr{C}^{Tr}$, *there is a strong correlation between* a *and* y, *s.t.* $\Pr(a = y) = p_{spu}$, *where* $1 \geq p_{spu} > 1/2$.

Recall from Section B.1 that the inputs in each modality are generated based on feature vectors. In SL, where we have only one modality, the situation becomes equivalent to the one analyzed in (Sagawa et al., 2020). Similar variants are studied in (Wald et al., 2021; Aubin et al., 2021; Yao et al., 2022) to investigate distribution shift and out-of-domain generalization. Despite its simplicity, this setup reflects key aspects of general distribution shift. Here, z_{core} is the core feature and z_{spu} is the spurious feature, such as 'grass' in the aforementioned example, or texture/backgrounds in ImageNet.

We assume that the core feature has a larger variance than the spurious feature, indicated by the values of σ_{core} and σ_{spu}. This is detailed in below, along with some additional assumptions.

Assumption 4 *The gap between the variances of the core and spurious features is significant:* $\sigma_{core} = \Theta(1)$, $\sigma_{core} \geq 1$ *and* $\sigma_{spu} = O(\frac{1}{\sqrt{\log n}})$. *The spurious correlation is large:* $p_{spu} = 1 - o(1)$. *We consider the high-dimensional (overparameterized) setting where* $n = \omega(1)$, $d_I = \Omega(n)$ *and* $d_T = \Omega(n)$. *The noise levels are not too large:* $\sigma_{\xi,I} = O(\log n)$ *and* $\sigma_{\xi,T} = O(\log n)$.

C.1.2 Comparing Robustness of SL and MMCL

Under Assumption 4, SL tends to associate labels mostly with the spurious feature, as they appear to be more stable and reliable for prediction compared to the core feature. This results in low accuracy when tested on the ground-truth distribution, as demonstrated in the following theorem.

Theorem 5 (Theorem 1 from (Sagawa et al., 2020)) *Let* $\boldsymbol{W}^*$ *represent the model trained using SL as described in Section B.3. Assuming that Assumption 4 holds, and* n *and* d_I *are sufficiently large, with a high probability, the accuracy of* $\boldsymbol{W}^*$ *on the true distribution satisfies* $Acc^{SL}_{\mathscr{P}^\star}(\boldsymbol{W}^*) \leq 2/3$. *Additionally, the model's test accuracy on examples where* $a \neq y$ *is* $\leq 1/3$, *worse than random chance.*

Next, we examine MMCL. From Lemma 2, we know that the similarity between an image with feature $\boldsymbol{z}$ and a text with feature $\boldsymbol{z}'$ is approximately $[z_{\text{core}}\ z_{\text{spu}}] \begin{bmatrix} 1+\sigma_{\text{core}}^2 & 2p_{\text{spu}}-1 \\ 2p_{\text{spu}}-1 & 1+\sigma_{\text{spu}}^2 \end{bmatrix} \begin{bmatrix} z'_{\text{core}} \\ z'_{\text{spu}} \end{bmatrix}$, showing that the variance of features ensures that image and text features, that share the underlying core features, have a higher similarity score. Furthermore, if we let $\boldsymbol{z}'$ be the feature $\bar{\boldsymbol{z}}_{y'} = [y' \ \ 0]^\top$ in label y''s corresponding prompt $\boldsymbol{p}_{y'}$, we deduce that the similarity to the prompt is approximately $(1+\sigma_{\text{core}}^2) y' z_{\text{core}} + (2p_{\text{spu}} - 1) y' z_{\text{spu}}$. Here, the core feature carries more weight when the variance is large. In essence, since the MMCL loss contrasts images and unpaired texts in the same latent classes, learning features that have high variance is encouraged; this is in contrast with SL, where features that have low variance are preferred. With the above observation, after bounding the effect of noise, we arrive at the following theorem (with proof in Appendix G.2).

Theorem 6 *Let $\boldsymbol{W}_I^*$ and $\boldsymbol{W}_T^*$ be the weights of the encoders trained using MMCL as described in Section B.2. Under Assumption 4* [1]*, with a high probability of at least $1 - O(\frac{1}{\text{poly}(n)}) = 1 - o(1)$, the encoders achieve the following zero-shot accuracy on the true distribution*

$$Acc_{\mathscr{P}^\star}^{MMCL}(\boldsymbol{W}_I^*, \boldsymbol{W}_T^*) \geq 1 - \frac{1}{2}\boldsymbol{\Phi}(\kappa_1) - \frac{1}{2}\boldsymbol{\Phi}(\kappa_2) - o(1),$$

where $\kappa_1 = \frac{2p_{spu} - 2 - \sigma_{core}^2}{\sqrt{(1+\sigma_{core}^2)^2\sigma_{core}^2 + (2p_{spu}-1)^2\sigma_{spu}^2}}$, $\kappa_2 = \frac{-2p_{spu} - \sigma_{core}^2}{\sqrt{(1+\sigma_{core}^2)^2\sigma_{core}^2 + (2p_{spu}-1)^2\sigma_{spu}^2}}$ and $\boldsymbol{\Phi}$ denotes the CDF of the standard normal distribution. Meanwhile, the model's test accuracy on examples where $a \neq y$ is lower bounded by $1 - \boldsymbol{\Phi}(\kappa_1) - o(1)$.

Corollary 7 *With $\sigma_{core} = 1$, for sufficiently large n, $Acc_{\mathscr{P}^\star}^{MMCL}(\boldsymbol{W}_I^*, \boldsymbol{W}_T^*) \geq 81\%$, Moreover, in this case, no model can achieve an accuracy higher than 85%.*

This, compared with Theorem 5, demonstrates that MMCL can outperform SL by a large margin, and comes close to achieving the best possible accuracy of 85%.

Additionally, in terms of performance on examples where the spurious correlation does not hold, i.e., $a \neq y$, it's evident that MMCL excels. As Theorem 5 shows, SL's accuracy is even worse than random chance. In contrast, Theorem 6 demonstrates that MMCL consistently performs better than random chance. It maintains random chance even in the worst-case scenario, as indicated by $\boldsymbol{\Phi}(\kappa_1) \leq \frac{1}{2}$, owing to $2p_{\text{spu}} - 2 - \sigma_{\text{core}}^2 \leq 0$. When $\sigma_{\text{core}} = 1$, it achieves an accuracy of 69%.

C.2 Robustness via Inter-class Feature Sharing

Next, consider the second scenario and demonstrate how MMCL benefits from annotated details in some latent classes to disassociate spurious correlations in other latent classes, while SL fails to grasp these details. For example, typical images of a 'tree' have green leaves. However, trees in in the background of images of 'wolf' or 'ski resort' may appear without leaves. SL, which only observes the labels, tends to overlook the trees without leaves as they do not contribute to learning 'wolf' and 'ski resort', thus incorrectly correlating trees with the color green. In contrast, in MMCL, if the trees without leaves are annotated in the captions, the model can disassociate the green leaves from tree.

C.2.1 Distribution of features

We first present the underlying feature distributions and then compare MMCL's robustness with SL.

Definition 8 (Data Model 2) ***True distribution $\mathscr{P}^\star$.*** *We have $2m$ latent classes in total, with labels $1, \ldots, 2m$. For each label y, we define a unique alias (k, c): $k = \lfloor (y+1)/2 \rfloor$, and $c = 1$ if y is odd, and $c = -1$ if y is even. The label is sampled uniformly. Let $\beta \in [0, 1)$. Given a label alias (k, c), the corresponding feature vector $\boldsymbol{z} = [z_1, z_2, \ldots, z_{2m}]^\top$ is generated as:*

$$\begin{array}{llll} \forall j \leq m, & \text{if } j = k \text{ then } z_j = c & \text{if } j \neq k \text{ then } z_j \sim U(\{-\beta, +\beta\}) \\ \forall j > m, & \text{if } j = k + m \text{ then } z_j \sim U(\{-\alpha, +\alpha\}) & \text{if } j \neq k + m \text{ then } z_j \sim U(\{-\beta\alpha, \beta\alpha\}) \end{array}$$

1. The theorem holds under more relaxed assumptions about the variances and spurious correlation level; see details in Appendix G.2), but here we use Assumption 4 to keep consistency with Theorem 5

where $U(S)$ denotes the uniform distribution over set S.

Training distribution $\mathscr{P}^{Tr}$. *The training distribution is similar to the true distribution, but with z_{k+m} always equal to $c\alpha$, making it appear as if the $k+m$-th coordinate also indicates the label.*

Here, each feature vector in latent class (k, c) (e.g., 'tree') has a core feature at coordinate k (characteristics of a tree) and a spurious feature at coordinate $k + m$ that correlates with the latent class in the training distribution but not the true distribution (e.g., the color green). With a large α, such a spurious feature has a larger magnitude than the true feature, making it easier to be learned. There are also other features at different coordinates that do not correlate with the label; these features are weaker (indicated by $\beta < 1$) so that they do not change the latent class. One observation is that examples in latent class other than (k, c) would show no correlation between the k-th and $k + m$-th features , hinting at their independence from each other (e.g., trees are not necessarily green). We will show that unlike SL, MMCL can leverage such a hint to obtain a superior robustness.

C.2.2 Comparing Robustness of SL and MMCL

The theorem below demonstrates that SL achieves a low accuracy under distribution shift when the spurious feature is strong, i.e., when α is large.

Theorem 9 *Assuming that the input noise in each modality is zero, i.e., $\sigma_{\xi,I} = \sigma_{\xi,T} = 0$, and all possible feature vectors in $\mathscr{P}^{Tr}$ uniformly appear in the training dataset.* [2] *Let $\boldsymbol{W}^*$ be the model trained using SL as described in Section B.3. The accuracy on the true distribution has the following upper bound: $Acc^{SL}_{\mathscr{P}^\star}(\boldsymbol{W}^*) \leq 50\% + \frac{2}{(1+\alpha^2)(1-\beta)^2-8}$. For example, if $\alpha = 10$ and $\beta = 1/3$, then $Acc^{SL}_{\mathscr{P}^\star}(\boldsymbol{W}^*) \leq 60\%$.*

Next, we will examine how MMCL leverages the information about independence of core and spurious features in each latent class, which is hidden in other latent classes. First, recall the conclusion in Lemma 2, and obtain that the similarity between an image with features $\boldsymbol{z}$ and a text with features $\boldsymbol{z}'$ is given by $\boldsymbol{z}^\top \begin{bmatrix} \frac{1+(m-1)\beta^2}{m}\mathbf{I}_m & \frac{\alpha}{m}\mathbf{I}_m \\ \frac{\alpha}{m}\mathbf{I}_m & \frac{1+(m-1)\beta^2}{m}\alpha^2\mathbf{I}_m \end{bmatrix} \boldsymbol{z}'$. The fact that β only appears on the diagonal and not on off-diagonal elements shows that the occurrence of core feature of a given latent class in other latent classes increases the weight for same-feature products between images and texts rather than different-feature products. For example, trees without green leaves in classes other than tree increase the covariance between texts and images of tree, but do not contribute to the correlation between tree and green. Hence appearance of green in any image has a limited impact on its similarity to a text describing a tree. More precisely, when computing the similarity between a given image and the prompt for a tree, a weight $\frac{1+(m-1)\beta^2}{m}$ is assigned to 'the true characteristic of a tree' and a weight $\frac{\alpha}{m}$ is assigned to 'green'. Here, a larger β leads to more weight placed on the core feature, highlighting how MMCL utilizes shared features between classes to enhance robustness. This insight leads us to the following theorem demonstrating the superior performance of MMCL under distribution shift.

2. Assumptions are made to simplify the analysis, but our analysis can be readily extended to show that same conclusions holds with high probability in broader settings with sufficient sample size and reasonable noise level.

Theorem 10 *Under the same assumption as in Theorem 9 Let $\boldsymbol{W}_I^*$ and $\boldsymbol{W}_T^*$ be the weights of encoders trained using MMCL as described in Section B.2. Then as long as $\beta^2 m > \frac{\alpha^2(1+\beta)}{1-\beta} - 1 + \beta^2$, the model has 100% zero-shot accuracy on the true distribution, i.e., $Acc_{\mathscr{D}^\star}^{MMCL}(\boldsymbol{W}_I^*, \boldsymbol{W}_T^*) = 100\%$.*

We also observe that if the features were not shared between classes, i.e., $\beta = 0$, it would be impossible for the model to achieve such performance. This once again emphasizes the role of shared features.

Important Consideration about Robustness An important question is whether the improvement in accuracy under distribution shift is solely due to MMCL's improvement in in-distribution generalization. In Appendix I, we demonstrate that we control for in-distribution generalization in both theoretical examples. Specifically, in Data Model 1, SL has slightly better in-distribution accuracy, while in Data Model 2, both SL and MMCL achieve 100% in-distribution accuracy. Thus, MMCL's improvement solely results from enhanced robustness, and in fact, both relative and effective robustness as defined in Taori et al. (2020).

Appendix D. Understanding the benefit of rich image captions

In Section C, we assumed that both $\boldsymbol{\mu}_I(\cdot)$ and $\boldsymbol{\mu}_T(\cdot)$ are identity, implying that the captions mentioned everything depicted in the image. However, in practice, captions often serve as annotations or illustrations accompanying the image, with certain details omitted. Empirical evidence suggests that rich captions are generally beneficial (Santurkar et al., 2022; Nguyen et al., 2023), but it remains unclear if richness of captions can affect robustness and, if so, how. In this section, we theoretically investigate this question by varying how much and what information is mentioned in captions. Specifically, we keep $\boldsymbol{\mu}_I(\cdot)$ as an identity function, while let $\boldsymbol{\mu}_T(\boldsymbol{z})$ represent a masked version of the original feature vector $\boldsymbol{z}$, where some information may not be reflected in caption.

Benefits of mentioning variations in the core features. Recall that in Section C.1, utilizing Data Model 1 (Definition 3), we showed that MMCL can learn large-variance core features better than SL, resulting in less reliance on the spurious feature. Now, we use the same data model to explore what happens if the feature variance is not fully reflected in the captions. For example, when the caption only contains the word 'cows' or 'grass', without describing their appearance.

Definition 11 (Feature masking in data model 1 (Definition 3)) *Given a feature vector $\boldsymbol{z} = \begin{bmatrix} z_{core} \\ z_{spu} \end{bmatrix}$ with corresponding y and a, we let $\boldsymbol{\mu}_T(\boldsymbol{z}) = \begin{bmatrix} y + \psi_{core}(z_{core} - y) \\ a + \psi_{spu}(z_{spu} - a) \end{bmatrix}$, with ψ_{core} drawn from Bernoulli(π_{core}) and ψ_{spu} drawn from Bernoulli(π_{spu}). Both π_{core} and π_{spu} are $\in [0, 1]$.*

Here, $z_{\text{core}} - y$ and $z_{\text{spu}} - a$ represent the variations in the core and spurious features, both of which are Gaussian random variables by Definition 3. This implies that the captions capture these variations with probabilities π_{core} and π_{spu}, respectively. When $\pi = 0$, caption ignores all the details and treats all features of the same kind as a single entity. The following theorem shows the effects of π_{core}, π_{spu}.

Theorem 12 *With data from data model 1 and $\boldsymbol{\mu}_T$ defined in Definition 11, with a high probability, the model trained using MMCL has a test accuracy on examples where the spurious*

correlation does not hold (i.e., $a \neq y$) given by $1 - \mathbf{\Phi}(\frac{2p-2-\pi_{core}^2\sigma_{core}^2}{\sqrt{(1+\pi_{core}^2\sigma_{core}^2)^2\sigma_{core}^2+(2p-1)^2\sigma_{spu}^2}}) \pm o(1)$. *The non-negligible part of this accuracy increases as π_{core} increases and is independent of π_{spu}.*

The theorem reveals that the model exhibits less reliance on the spurious correlation when the caption mentions the variance in the core feature (e.g., appearance of the cow in each specific image). Additionally, we notice that mentioning variance in the spurious feature has minimal effect on the robustness, as it does not impact the correlation with the core feature.

Mentioning more features benefits robustness. Next, we utilize data model 2 to explore the effect of mentioning more features in the captions.

Definition 13 (Feature masking in data model 2 (Definition 8)) *For a feature vector $\boldsymbol{z}$ with label (k, c), let $\boldsymbol{\mu}_T(\boldsymbol{z}) = \boldsymbol{\psi} \odot \boldsymbol{z}$, where $\boldsymbol{\psi} = [\psi_1 \dots \psi_l]^\top$ with $\psi_k = 1$ and $\psi_j \sim$ Bernoulli(π) for $j \neq k$.*

Here, the caption always mentions the feature indicating the latent class, while other features are mentioned with a probability π. Note that $\pi = 0$ corresponds to the setting where the caption is just the same as the label. The following theorem demonstrates that the model can achieve robustness only when the caption sufficiently mentions features that are not directly related to the image's latent class.

Theorem 14 *With data model 2 and $\boldsymbol{\mu}_T$ defined in Definition 13, let $\boldsymbol{W}_I^*$ and $\boldsymbol{W}_T^*$ be the weights of encoders trained using MMCL. Then the model's accuracy on the true distribution satisfies $Acc_{\mathscr{P}^\star}^{MMCL}(\boldsymbol{W}_I^*, \boldsymbol{W}_T^*) = 100\%$ if $\pi > \tilde{\pi}$, and $Acc_{\mathscr{P}^\star}^{MMCL}(\boldsymbol{W}_I^*, \boldsymbol{W}_T^*) \leq 50\%$ if $\pi < \tilde{\pi}$, where $\tilde{\pi} := \frac{(1+\beta)\alpha^2-1+\beta}{(1-\beta)\beta^2(m-1)}$.*

As explained in Section C.2, even if certain features do not directly indicate labels for a class, they can still help learn relationships between features (for example, not all trees are green), and this knowledge can be valuable for other classes. However, if these features are missing from the captions, they contribute less to the cross-covariance matrix used by the model for predictions (Lemma 2). In the extreme case where $\pi = 0$, captions reduce to labels used by SL, and robustness does not improve.

Appendix E. Experiments

In this section, we perform experiments to demonstrate the crucial role of the MMCL loss function and the richness of captions in robustness, validating our theoretical results.

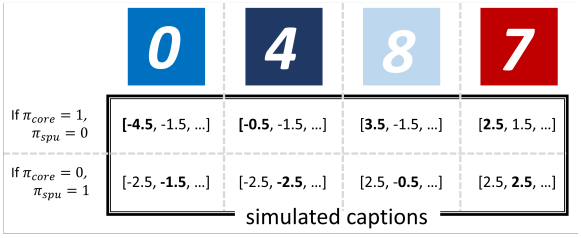
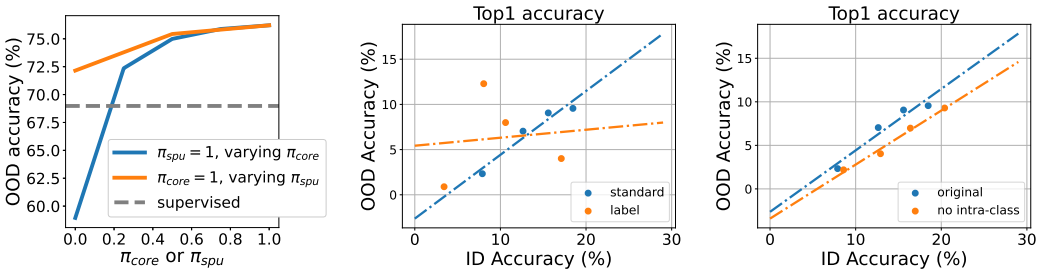


Figure 1: Construction of captions.

E.1 A Semi-synthetic Experiment

We conduct a carefully designed semi-synthetic binary classification experiment to showcase MMCL's robustness and the significance of rich captions. The task is to distinguish digits 0 to 4 (class 1) from digits 5 to 9 (class 2). In the training set, MNIST (Deng, 2012) digits are placed on colored



(a) (b) (c)

Figure 2: (a) OOD accuracy on the semi-synthetic data. A large π_{core} is crucial for ensuring MMCL's superior robustness compared to SL, but the value of π_{spu} has minimal effect. (b) and (c) display OOD-ID relation on MS COCO. (b) compares of robustness between two ways of captioning image. (c) demonstrates the effect of intra-class contrasting on the robustness of the trained model.

backgrounds, including three types of blue and three types of red. As illustrated Figure 6, for digits 0-4, 99.5% of images have randomly selected shades of blue as the background, while the remaining 0.5% have random red backgrounds. The same applies to digits 5-9, but with blue and red swapped. In the test set, backgrounds are randomly chosen for all images. Therefore, digits represent the core feature, while colors serve as the spurious feature whose correlation with classes only exist in the training data. Captions are simulated as vectors, where the first coordinate contains digit information and the second contains color information.

Both features exhibit variance; for example, there are four variations of digits between 0 and 4 and three variations of blue backgrounds. We use π_{core} and π_{spu} to control the specificity of the captions, determining how much the caption mentions the variance in each feature. Being 'specific' means mentioning the exact value (e.g., specifying a particular shade of blue), while 'not specific' means referring to a value that represents an entire category (e.g., using the mean value for three shades of blue to represent any blue). Figure 1 shows an example. For more details, please refer to Appendix J.1.

We plot the accuracy on the out-of-distribution test set in Figure 2a, while varying the values of π_{core} and π_{spu}. We observe: (1) With sufficiently rich captions (high π_{core}), MMCL exhibits better robustness than SL (horizontal line). (2) A high π_{core}, indicating that the captions mentioning the variance of the core feature, is essential for achieving robustness, as reducing π_{core} significantly hurts the robustness. (3) In contrast, π_{spu} has minimal effect on robustness. It's worth noting that (2) and (3) directly validate the conclusions from Theorem 12. Additional discussion can be found in Appendix J.1.

E.2 Experiments on MS COCO

We compare models trained with MMCL and with SL on MSCOCO dataset. For MMCL, we use the popular CLIP loss, and for SL we use Cross-Entropy loss. We use ResNet50 as the image encoder and Transformer as the text encoder. The test datasets include the original MSCOCO testset (in-distribution), and six different versions of (shifted) ImageNet

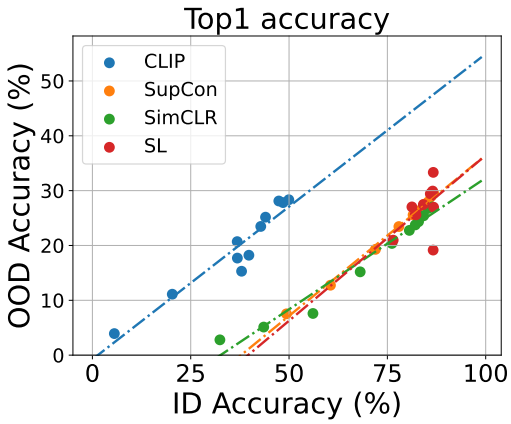


Figure 3: Comparison of robustness (OOD-ID relation) on CC3M between different learning approaches when changing the model width.

(out-of-distribution) Hendrycks et al. (2021a,b); Recht et al. (2019); Wang et al. (2019); Russakovsky et al. (2015b); Barbu et al. (2019). Other details are in Appendix J.2.

Richness of captions is critical in achieving robustness. To demonstrate the importance of richness in robustness, we train another version of CLIP where the captions are reduced to be the same as the labels. As shown in Figure 2b, this results in reduced robustness. It emphasizes the significance of including additional details in captions beyond just the label information, confirming the conclusions drawn from our theoretical analysis in Section D. Additional results of the top5 accuracy for the zero-shot classification task in Appendix J.2 also show the same trend.

Effect of intra-class contrasting. To demonstrate the mechanism theoretically shown in Section C.1, we consider a modified CLIP loss where we drop pairs from the denominator if they belong to the same class. This leads to no contrasting between images and texts from the same class. As illustrated in Figure 2c, we observe that reducing intra-class contrasting, leads to reduced robustness. The results of the top5 accuracy in Appendix J.2 also demonstrate the same trend.

E.3 Experiments on Conceptual Captions

Comparison between different learning approaches. In this experiment, we compare CLIP with SL, SupCon, and SimCLR. The model width (i.e., the hidden and output dimensions of MLP) varies in the set $\{16, 32, 64, 128, 256, 512, 768, 1024, 2048, 4096, 8192\}$. Figure 3 illustrates that training CLIP yields better robustness than other learning approaches in terms of top 1 accuracy. This result is well aligned with our analysis in Appendix K. Interestingly, the robustness of SL, SupCon, and SimCLR is almost similar, resulting in nearly identical slopes.

Richness of captions. Similar to the experiments on MSCOCO, we train another version of CLIP where the captions are reduced to be the same as the class labels. We also train with different MLP widths from 16 to 8192. Figure 4 illustrates that training CLIP on the standard captions of CC3M yields better robustness than training on the reduced

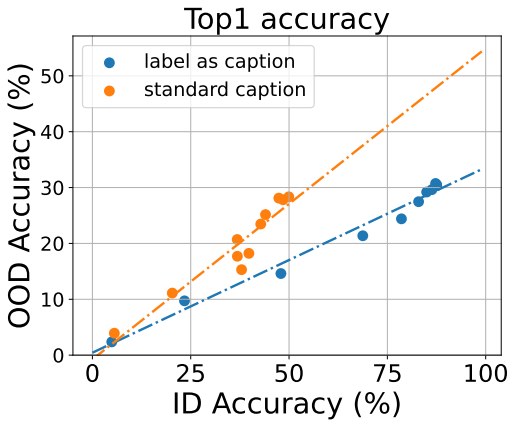


Figure 4: Comparison of robustness (OOD-ID relation) on CC3M between two ways of captioning images when changing the model width.

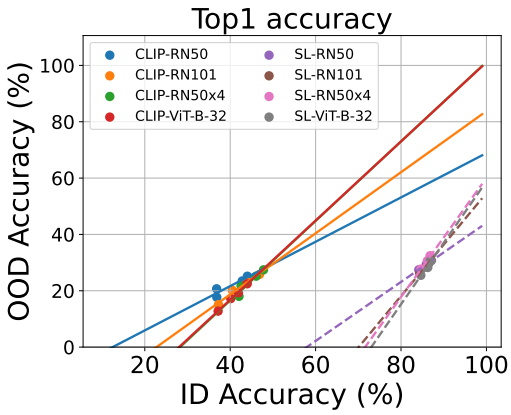


Figure 5: Comparison of robustness (OOD-ID relation) on CC3M between CLIP and SL when changing the image encoder's architecture. The solid lines represent CLIP models while the dashed ones represent SL models.

captions. This finding again reinforces our theoretical results about the benefit of rich image captions in Section D.

Comparison between CLIP and SL with different image encoder's architectures. In this experiment, we change the architecture of the CLIP image encoder. We utilize four different architectures including ResNet50, ResNet101, ResNet50x4, and ViT-B-32 whose pre-trained weights are available. Figure 5 illustrates that training CLIP on CC3M yields better robustness than training SL across different encoders' architectures.

Appendix F. Preliminaries

F.1 Minimizer of MMCL loss

Nakada et al. (2023) has shown the equivalence between minimizing linear MMCL and SVD. We reiterate this for reference in our proof. As defined in Section B.2, our loss function is

$$\mathcal{L}_{\mathrm{MMCL}}(\boldsymbol{W}_I,\boldsymbol{W}_T)=\frac{1}{2n(n-1)}\sum_i\sum_{j\neq i}(s_{ij}-s_{ii})+\frac{1}{2n(n-1)}\sum_i\sum_{j\neq i}(s_{ji}-s_{ii})+\frac{\rho}{2}\|\boldsymbol{W}_I^\top\boldsymbol{W}_T\|_F^2,$$

which can be rewritten as a matrix factorization objective

$$\mathcal{L}(\boldsymbol{W}_I,\boldsymbol{W}_T)=-\operatorname{Tr}(\boldsymbol{W}_I\boldsymbol{S}\boldsymbol{W}_T^\top)+\frac{\rho}{2}\|\boldsymbol{W}_I^\top\boldsymbol{W}_T\|_F^2, \tag{2}$$

where $\boldsymbol{S}$ is the cross-covariance matrix

$$\boldsymbol{S}\coloneqq\frac{1}{n}\sum_i\boldsymbol{x}_{I,i}\boldsymbol{x}_{T,i}^\top-\frac{1}{n(n-1)}\sum_{i\neq j}\boldsymbol{x}_{I,i}\boldsymbol{x}_{T,j}^\top. \tag{3}$$

The following directly follows from Eckart-Young-Mirsky theorem: let $\sum_{i=1}^d\lambda_i\boldsymbol{u}_{I,i}\boldsymbol{u}_{T,i}^\top$ with $\lambda_1\geq\lambda_2\geq\ldots\lambda_d>0$ be the SVD of $\boldsymbol{S}$, and let $\boldsymbol{W}_I^*,\boldsymbol{W}_T^*$ be the minimizer of the loss, then

$$\boldsymbol{W}_I^{*\top}\boldsymbol{W}_T^*=\frac{1}{\rho}\sum_{i=1}^p\lambda_i\boldsymbol{u}_{I,i}\boldsymbol{u}_{T,i}^\top \tag{4}$$

F.2 Minimizer of SL loss

We note that in both Data Model 1 and Data Model 2, under the assumptions we've made, the training data are separable, or separable with high probability. As shown in Soudry et al. (2018), minimizing the logistic loss or Cross-Entropy losses with a linear model at a sufficiently small step size converges to the solution for a hard-margin SVM. Therefore, in our analysis for SL, we equivalently examine this solution.

Appendix G. Analysis for Data Model 1

G.1 Analysis for SL

Here, we briefly explain how Theorem 6 is derived from Theorem 1 in (Sagawa et al., 2020). Firstly, we provide the following version, which is a direct translation from (Sagawa et al., 2020) but with our notations. First let n_{maj} denote the number of examples with $a=y$ in the training set and n_{maj} denote the number of examples with $a\neq y$ in the training set. Define $\sigma_{\mathrm{core}}'^2\coloneqq\sigma_{\mathrm{core}}^2+\frac{\sigma_{\xi,I}^2}{d_I}$, $\sigma_{\mathrm{spu}}'^2\coloneqq\sigma_{\mathrm{spu}}^2+\frac{\sigma_{\xi,I}^2}{d_I}$, $\sigma_{\xi,I}'^2\coloneqq\frac{\sigma_{\xi,I}^2(d-2)}{d}$.

Theorem 15 *For any $\frac{n_{maj}}{n}\geq1-\frac{1}{2001}$, $\sigma_{core}'^2\geq1$, $\sigma_{spu}'^2\leq\frac{1}{16\log100n_{maj}}$, $\sigma_{\xi,I}'^2\leq\frac{n_{maj}}{600^2}$ and $n_{min}\geq100$, there exists d_0 such that for all $d>d_0$, with high probability over draws of the training data*

Test error on examples where $a\neq y$ achieved by $\boldsymbol{W}^*\geq2/3$.

It is now easy to see that the quantities in Assumption 4 can satisfy the conditions in the theorem above when they become sufficiently asymptotic. Note that the above statement is about test error on examples where $a \neq y$, which accounts for half of the entire distribution $\mathscr{P}^\star$, so the accuracy on the entire distribution $\mathscr{P}^\star$ is at most $\frac{1}{2} \times 100\% + \frac{1}{2} \times (1 - 2/3) = 2/3$.

G.2 Analysis for MMCL

We note that Theorem 6 holds under a more relaxed assumption, which the following anlaysis is based on.

Assumption 16 *The gap between the variances of the core and spurious features is significant: $\sigma_{core} - \sigma_{spu} = \Theta(1)$ and $\sigma_{spu} = O(1)$. p can be any value between $\frac{1}{2}$ and 1. We consider the high-dimensional (overparameterized) setting where $n = \omega(1)$, $d_I = \Omega(n)$ and $d_T = \Omega(n)$. The noise levels are not too large: $\sigma_{\xi,I} = O(\log n)$ and $\sigma_{\xi,T} = O(\log n)$.*

We define the following notations. We write each $\boldsymbol{z_i}$ as $\boldsymbol{z_i} = \begin{bmatrix} y_i + \zeta_{1,i} \\ a_i + \zeta_{2,i} \end{bmatrix}$, where $\zeta_{1,i}$ and $\zeta_{2,i}$ are Gaussian variables according to our definition. We also let $\boldsymbol{\zeta}_i = \begin{bmatrix} \zeta_{1,i} \\ \zeta_{2,i} \end{bmatrix}$. Let $\mathfrak{Z} = [\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \ldots, \boldsymbol{\zeta}_n]$, $\boldsymbol{\Xi}_I = [\boldsymbol{\xi}_{I,1}, \boldsymbol{\xi}_{I,2}, \ldots, \boldsymbol{\xi}_{I,n}]$, $\boldsymbol{\Xi}_T = [\boldsymbol{\xi}_{T,1}, \boldsymbol{\xi}_{T,2}, \ldots, \boldsymbol{\xi}_{T,n}]$. Additionally, we let $\boldsymbol{F} = \begin{bmatrix} y_1, y_2, \ldots, y_n \\ a_1, a_2, \ldots, a_n \end{bmatrix}$.

G.2.1 Useful Concentration Bounds

Lemma 17 (Montgomery-Smith (1990)) *Let $\{x_i\}_{i=1}^n$ be a set of random Rademacher variables, then with probability at least $1 - \delta$, the following holds*

$$|\frac{1}{n}\sum_{i=1}^{n} x_i| \leq \sqrt{\frac{2\ln(1/\delta)}{n}}$$

Lemma 18 (Mills' ratio. Exercise 6.1 in (Shorack and Shorack, 2000).) *Let v be a Gaussian random variable drawn from $\mathcal{N}(0,1)$. Then for all $\lambda > 0$,*

$$\frac{\lambda}{\lambda^2+1}\frac{1}{\sqrt{2\pi}}e^{-\frac{\lambda^2}{2}} < \Pr(v \geq \lambda) < \frac{1}{\lambda}\frac{1}{\sqrt{2\pi}}e^{-\frac{\lambda^2}{2}}.$$

Corollary 19 *Let $\{x_i\}_{i=1}^n$ be a set of random variables independently drawn from $\mathcal{N}(0,\sigma^2)$, then with probability at least $1 - O(\frac{1}{\text{poly}(n)})$, the following holds*

$$|\frac{1}{n}\sum_{i=1}^{n} x_i| \leq O(\sigma\sqrt{\frac{\log n}{n}})$$

Lemma 20 (Restatement of Theorem 6.1 from (Wainwright, 2019)) *Let $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ be a random matrix with $\boldsymbol{x}_i^\top$ as its i-th row. Each $\boldsymbol{x}_i$ is drawn i.i.d. from $\mathcal{N}(0,\boldsymbol{\Sigma})$. Then with probability at least $1-\delta$, the maximal singular value of $\boldsymbol{X}$, denoted as $\sigma_{\max}(\boldsymbol{X})$ satisfies the following*

$$\sigma_{\max}(\boldsymbol{X}) \leq \sqrt{n}\gamma_{\max}(\sqrt{\boldsymbol{\Sigma}})(1 + \frac{\sqrt{2\ln(1/\delta)}}{n}) + \sqrt{\text{Tr}(\boldsymbol{\Sigma})}.$$

Lemma 21 (Gaussian covariance estimation, Example 6.3 from (Wainwright, 2019)) *Let* $\{\boldsymbol{v}_i\}_{i=1}^n$ *be a set of vectors* $\in \mathbb{R}^d$ *independently drawn from* $\mathcal{N}(0, \boldsymbol{\Sigma})$*, and let* $\hat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{v}_i \boldsymbol{v}_i^\top$ *then with probability at least* $1 - 2e^{-n\delta^2/2}$

$$\frac{\|\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}\|_2}{\|\boldsymbol{\Sigma}\|_2} \leq 2\sqrt{\frac{d}{n}} + 2\delta + (\sqrt{\frac{d}{n}} + 2)^2.$$

G.2.2 Concentration of the Cross-covariance

Concentrations in Low-dimensional Underlying Feature Space:

Lemma 22 *With probability at least* $1 - \delta$ *the following holds*

$$|\frac{1}{n}\sum_{i=1}^n a_i y_i - (2p-1)| \leq \sqrt{2\frac{\ln(1/\delta)}{n}}.$$

Proof By Hoeffding's inequality. ■

Lemma 23 *With probability at least* $1 - O(\frac{1}{\text{poly}(n)})$ *the following holds*

$$\|\frac{1}{n}\sum_{i=1}^n \boldsymbol{\zeta}_i \boldsymbol{\zeta}_i^\top - \begin{bmatrix} \sigma_{core}^2 & 0 \\ 0 & \sigma_{spu}^2 \end{bmatrix}\|_2 \leq O(\sigma_{core}^2\sqrt{\frac{\log n}{n}})$$

Proof By Lemma 21. ■

Lemma 24 *With probability at least* $1 - O(\frac{1}{\text{poly}(n)})$

$$\|\frac{1}{n}\sum_{i=1}^n \begin{bmatrix} y_i \\ a_i \end{bmatrix} \boldsymbol{\zeta}_i^\top\|_2 \leq O(\sigma_{core}\sqrt{\frac{\log n}{n}})$$

Proof By Lemma 19. ■

Lemma 25 *With probability at least* $1 - O(\frac{1}{\text{poly}(n)})$

$$\|\frac{1}{n}\sum_{i=1}^n \boldsymbol{z}_i \boldsymbol{z}_i^\top - \begin{bmatrix} 1+\sigma_{core}^2 & 2p-1 \\ 2p-1 & 1+\sigma_{spu}^2 \end{bmatrix}\|_2 \leq O(\sqrt{\frac{\log n}{n}}) \tag{5}$$

Proof Write $\frac{1}{n}\sum_{i=1}^n \boldsymbol{z}_i \boldsymbol{z}_i^\top$ as the following

$$\frac{1}{n}\sum_{i=1}^n \boldsymbol{z}_i \boldsymbol{z}_i^\top = \frac{1}{n}\sum_{i=1}^n \begin{bmatrix} y_i^2 & y_i a_i \\ a_i y_i & a_i^2 \end{bmatrix} + \frac{1}{n}\sum_{i=1}^n \begin{bmatrix} y_i \\ a_i \end{bmatrix} \boldsymbol{\zeta}_i^\top + \frac{1}{n}\sum_{i=1}^n \boldsymbol{\zeta}_i [y_i \quad a_i] + \frac{1}{n}\sum_{i=1}^n \boldsymbol{\zeta}_i \boldsymbol{\zeta}_i^\top.$$

Then invoking Lemmas 22, 23 and 24 completes the proof. ■

Lemma 26 *With probability at least* $1 - O(\frac{1}{\text{poly}(n)})$

$$\|\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{z}_i\|_2 \leq O(\sqrt{\frac{\log n}{n}})$$

Proof Note that each of the two elements in $\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{z}_i$ can be seen as the sum of the mean of n independent Rademacher variables and the mean of n independent Gaussian variables. Combining Lemmas 17 and 19 completes the proof. ■

Lemma 27 *With probability at least* $1 - O(\frac{1}{\text{poly}(n)})$

$$\left\|\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{z}_i\boldsymbol{z}_i^\top - \frac{1}{n(n-1)}\sum_{i\neq j}^{n} \boldsymbol{z}_i\boldsymbol{z}_j^\top - \begin{bmatrix} 1+\sigma_{core}^2 & 2p-1 \\ 2p-1 & 1+\sigma_{spu}^2 \end{bmatrix}\right\|_2 = O(\sqrt{\frac{\log n}{n}}).$$

Proof

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{z}_i\boldsymbol{z}_i^\top - \frac{1}{n(n-1)}\sum_{i\neq j}^{n} \boldsymbol{z}_i\boldsymbol{z}_j^\top =& \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{z}_i\boldsymbol{z}_i^\top - \frac{1}{n(n-1)}\sum_{i=1}^{n} \boldsymbol{z}_i\sum_{j=1}^{n} \boldsymbol{z}_j + \frac{1}{n(n-1)}\sum_{i=1}^{n} \boldsymbol{z}_i\boldsymbol{z}_i^\top \\ =& \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{z}_i\boldsymbol{z}_i^\top - \frac{n}{n-1}\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{z}_i\frac{1}{n}\sum_{j=1}^{n} \boldsymbol{z}_j + \frac{1}{n(n-1)}\sum_{i=1}^{n} \boldsymbol{z}_i\boldsymbol{z}_i^\top.\end{aligned}$$

Note that by Lemma 26, the norm of the second term on the RHS is $O(\frac{\log n}{n})$, and by Lemma 25 the norm of the third term is $O(\frac{1}{n})$. Combining these results and applying Lemma 25 to the first term yields:

$$\left\|\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{z}_i\boldsymbol{z}_i^\top - \frac{1}{n(n-1)}\sum_{i\neq j}^{n} \boldsymbol{z}_i\boldsymbol{z}_j^\top - \begin{bmatrix} 1+\sigma_{\text{core}}^2 & 2p-1 \\ 2p-1 & 1+\sigma_{\text{spu}}^2 \end{bmatrix}\right\|_2 = O(\sqrt{\frac{\log n}{n}}).$$

■

Concentrations in High-dimensional Input Space:

Corollary 28 *By applying Lemma 20, we can conclude that the following statements hold with a probability of at least* $1 - 3\delta$

$$\begin{aligned}\|\boldsymbol{\beta}\|_2 \leq& \sqrt{n}\sigma_{core}(1 + \frac{\sqrt{2\ln(1/\delta)}}{n}) + \sqrt{\sigma_{core}^2 + \sigma_{spu}^2} \\ \|\boldsymbol{\Xi}_I\|_2 \leq& \sigma_{\xi,I}(\sqrt{\frac{n}{d_I}} + \sqrt{\frac{2\ln(1/\delta)}{nd_I}}) + \sigma_{\xi,I} \\ \|\boldsymbol{\Xi}_T\|_2 \leq& \sigma_{\xi,T}(\sqrt{\frac{n}{d_T}} + \sqrt{\frac{2\ln(1/\delta)}{nd_T}}) + \sigma_{\xi,T}\end{aligned}$$

Lemma 29 *With probability at least* $1 - O(\frac{1}{\text{poly}(n)})$

$$\|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\xi}_{I,i}\| \leq O(\frac{\sigma_{\xi,I}}{\sqrt{n}}) \quad \textit{and} \quad \|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\xi}_{T,i}\| \leq O(\frac{\sigma_{\xi,T}}{\sqrt{n}}).$$

Proof This can be obtained by recognizing that $\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\xi}_{I,i}$ can be treated as a single sample from $\mathcal{N}(0, \frac{\sigma_I^2}{nd_I}\mathbf{I}_{d_I})$ and by applying Lemma 20 with $\delta = \frac{1}{\text{poly}(n)}$. A similar argument applies to $\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\xi}_{T,i}$. ■

Lemma 30 *With probability at least* $1 - O(\frac{1}{\text{poly}(n)})$

$$\begin{aligned}\left\|\boldsymbol{S} - \boldsymbol{D}_I\begin{bmatrix}1+\sigma_{core}^2 & 2p-1\\ 2p-1 & 1+\sigma_{spu}^2\end{bmatrix}\boldsymbol{D}_T^\top\right\|_2 \leq & O(\frac{\sigma_{\xi,T}+\sigma_{\xi,I}}{\sqrt{n}} + \frac{\sigma_{\xi,I}\sigma_{\xi,T}}{n})\\ & + O(\frac{\sigma_{\xi,A}\sqrt{\log n}+\sigma_{\xi,B}\sqrt{\log n}+\sigma_{\xi,I}\sigma_{\xi,T}}{n})\\ & + O(\sqrt{\frac{\log n}{n}}),\end{aligned}$$

where $\boldsymbol{S}$ *is defined in Equation 3.*

Proof Firstly let's write $\boldsymbol{S}$ as

$$\begin{aligned}\boldsymbol{S} =& \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_{I,i}\boldsymbol{x}_{T,i}^\top - \frac{1}{n(n-1)}\sum_{i\neq j}\boldsymbol{x}_{I,i}\boldsymbol{x}_{T,i}^\top \\ =& \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{D}_I\boldsymbol{z}_i\boldsymbol{z}_i^\top\boldsymbol{D}_T^\top - \frac{1}{n(n-1)}\sum_{i\neq j}\boldsymbol{D}_I\boldsymbol{z}_i\boldsymbol{z}_j^\top\boldsymbol{D}_T^\top + \boldsymbol{R}\end{aligned} \tag{6}$$

where

$$\begin{aligned}\boldsymbol{R} =& \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{D}_I\boldsymbol{z}_i\boldsymbol{\xi}_{T,i}^\top + \boldsymbol{\xi}_{I,i}\boldsymbol{z}_i^\top\boldsymbol{D}_T^\top + \boldsymbol{\xi}_{I,i}\boldsymbol{\xi}_{T,i}^\top) - \frac{1}{n(n-1)}\sum_{i\neq j}(\boldsymbol{D}_I\boldsymbol{z}_i\boldsymbol{\xi}_{T,j}^\top + \boldsymbol{\xi}_{I,i}\boldsymbol{z}_j^\top\boldsymbol{D}_T^\top + \boldsymbol{\xi}_{I,i}\boldsymbol{\xi}_{T,j}^\top)\\ =& \underbrace{\frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{D}_I\boldsymbol{z}_i\boldsymbol{\xi}_{T,i}^\top + \boldsymbol{\xi}_{I,i}\boldsymbol{z}_i^\top\boldsymbol{D}_T^\top + \boldsymbol{\xi}_{I,i}\boldsymbol{\xi}_{T,i}^\top)}_{\boldsymbol{R}_1}\\ & \underbrace{- \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j=1}^{n}(\boldsymbol{D}_I\boldsymbol{z}_i\boldsymbol{\xi}_{T,j}^\top + \boldsymbol{\xi}_{I,i}\boldsymbol{z}_j^\top\boldsymbol{D}_T^\top + \boldsymbol{\xi}_{I,i}\boldsymbol{\xi}_{T,j}^\top)}_{\boldsymbol{R}_2}.\end{aligned}$$

Let's rewrite $\boldsymbol{R}_1$ as

$$\boldsymbol{R}_1 = \frac{1}{n-1}\Big(\boldsymbol{D}_I(\boldsymbol{F}+\boldsymbol{\mathfrak{Z}})\boldsymbol{\Xi}_T^\top + \boldsymbol{\Xi}_I(\boldsymbol{F}+\boldsymbol{\mathfrak{Z}})^\top\boldsymbol{D}_T^\top + \boldsymbol{\Xi}_I\boldsymbol{\Xi}_T^\top\Big).$$

Then

$$\|\boldsymbol{R}_1\|_2 \leq \frac{1}{n-1}\bigg(\|\boldsymbol{D}_I\|_2(\|\boldsymbol{F}\|_2+\|\boldsymbol{\mathfrak{Z}}\|_2)\|\boldsymbol{\Xi}_T\|_2+\|\boldsymbol{\Xi}_I\|_2(\|\boldsymbol{F}\|_2+\|\boldsymbol{\mathfrak{Z}}\|_2)\|\boldsymbol{D}_T\|_2+\|\boldsymbol{\Xi}_I\|_2\|\boldsymbol{\Xi}_T\|_2\bigg).$$

Note that $\|\boldsymbol{D}_I\|_2 = \|\boldsymbol{D}_T\|_2 = 1$ since they have orthonormal columns. Additionally, we can observe that $\|\boldsymbol{F}\|_2 \leq \|\boldsymbol{F}\|_F = \sqrt{2n}$. By combining these and applying Corollary 28 with $\delta = O(\frac{1}{\text{poly}(n)})$ we obtain the following

$$\begin{aligned}\|\boldsymbol{R}_1\|_2 \leq& O(\frac{\sigma_{\xi,T}+\sigma_{\text{core}}\sigma_{\xi,T}+\sigma_{\xi,I}+\sigma_{\text{core}}\sigma_{\xi,I}}{\sqrt{n}}+\frac{\sigma_{\xi,I}\sigma_{\xi,T}}{n})\\ =& O(\frac{\sigma_{\xi,T}+\sigma_{\xi,I}}{\sqrt{n}}+\frac{\sigma_{\xi,I}\sigma_{\xi,T}}{n})\end{aligned} \tag{7}$$

Next, we rewrite $\boldsymbol{R}_2$ as

$$\boldsymbol{R}_2 = \frac{n}{n-1}\bigg(\boldsymbol{D}_I\frac{1}{n}\sum_{i=1}^n \boldsymbol{z}_i(\frac{1}{n}\sum_{j=1}^n \boldsymbol{\xi}_{T,j})^\top + \frac{1}{n}\sum_{i=1}^n \boldsymbol{\xi}_{I,i}(\frac{1}{n}\sum_{j=1}^n \boldsymbol{z}_j)^\top \boldsymbol{D}_T^\top + \frac{1}{n}\sum_{i=1}^n \boldsymbol{\xi}_{I,i}(\frac{1}{n}\sum_{i=1}^n \boldsymbol{\xi}_{T,i})^\top\bigg).$$

Then applying Lemmas 26 and 29 yields

$$\|\boldsymbol{R}_2\|_2 \leq O(\frac{\sigma_{\xi,A}\sqrt{\log n}+\sigma_{\xi,B}\sqrt{\log n}+\sigma_{\xi,I}\sigma_{\xi,T}}{n}). \tag{8}$$

Additionally we observe that

$$\frac{1}{n}\sum_{i=1}^n \boldsymbol{D}_I\boldsymbol{z}_i\boldsymbol{z}_i^\top\boldsymbol{D}_T^\top - \frac{1}{n(n-1)}\sum_{i\neq j}\boldsymbol{D}_I\boldsymbol{z}_i\boldsymbol{z}_j^\top\boldsymbol{D}_T^\top = \boldsymbol{D}_I\left(\frac{1}{n}\sum_{i=1}^n \boldsymbol{z}_i\boldsymbol{z}_i^\top - \frac{1}{n(n-1)}\sum_{i\neq j}\boldsymbol{z}_i\boldsymbol{z}_j^\top\right)\boldsymbol{D}_T^\top.$$

Therefore by Lemma 25

$$\begin{aligned}&\left\|\frac{1}{n}\sum_{i=1}^n \boldsymbol{D}_I\boldsymbol{z}_i\boldsymbol{z}_i^\top\boldsymbol{D}_T^\top - \frac{1}{n(n-1)}\sum_{i\neq j}\boldsymbol{D}_I\boldsymbol{z}_i\boldsymbol{z}_j^\top\boldsymbol{D}_T^\top - \boldsymbol{D}_I\begin{bmatrix}1+\sigma_{\text{core}}^2 & 0\\ 0 & 1+\sigma_{\text{spu}}^2\end{bmatrix}\boldsymbol{D}_T^\top\right\|_2\\ \leq&\|\boldsymbol{D}_I\|_2\|\boldsymbol{D}_T\|_2 O(\sqrt{\frac{\log n}{n}})\\ =& O(\sqrt{\frac{\log n}{n}})\end{aligned} \tag{9}$$

Then, combining Equations 9, 7, 8, 6 yields

$$\begin{aligned}\left\|\boldsymbol{S}-\boldsymbol{D}_I\begin{bmatrix}1+\sigma_{\text{core}}^2 & 2p-1\\ 2p-1 & 1+\sigma_{\text{spu}}^2\end{bmatrix}\boldsymbol{D}_T^\top\right\|_2 \leq& O(\frac{\sigma_{\xi,T}+\sigma_{\xi,I}}{\sqrt{n}}+\frac{\sigma_{\xi,I}\sigma_{\xi,T}}{n})\\ &+O(\frac{\sigma_{\xi,A}\sqrt{\log n}+\sigma_{\xi,B}\sqrt{\log n}+\sigma_{\xi,I}\sigma_{\xi,T}}{n})\\ &+O(\sqrt{\frac{\log n}{n}})\end{aligned}$$

■

G.2.3 Perturbation in SVD

Lemma 31 *$\boldsymbol{G}^* \in \mathbb{R}^{d_I \times d_T}$ is a matrix whose SVD is $\sum_{i=1}^2 \lambda_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$ where $\lambda_1 > \lambda_2 > 0$, $\lambda_1 = \Theta(1)$, $\lambda_2 = \Theta(1)$ and $\lambda_1 - \lambda_2 = \Theta(1)$. $\boldsymbol{G} = \boldsymbol{G}^* + \boldsymbol{E}$ where $\|\boldsymbol{E}\|_2 \leq \epsilon$. Let $\sum_{i=1}^r \tilde{\lambda}_i \tilde{\boldsymbol{u}}_i \tilde{\boldsymbol{v}}_i^\top$ be the SVD of $\boldsymbol{G}$. If $\epsilon = o(1)$, then $\|\sum_{i=1}^p \tilde{\lambda}_i \tilde{\boldsymbol{u}}_i \tilde{\boldsymbol{v}}_i^\top - \boldsymbol{G}^*\|_2 \leq O(\sqrt{\epsilon})$, where $2 \leq p \leq r$.*

Proof Weyl's Theorem tells us that $|\tilde{\lambda_1} - \lambda_1| \leq \epsilon$, $|\tilde{\lambda_2} - \lambda_2| \leq \epsilon$ and $|\tilde{\lambda_i}| \leq \epsilon$ for $i \geq 3$. Now, let's define $\delta = \min\{\lambda_1 - \lambda_2, \lambda_2\}$. By applying Wedin's Theorem (Wedin, 1972) with the singular values partitioned into $\{\lambda_1\}$ and $\{\lambda_i\}_{i \neq 1}$, we have $\sin\theta(\boldsymbol{u}_1, \tilde{\boldsymbol{u}}_1) \leq O(\frac{\epsilon}{\delta})$. Similarly, by applying Wedin's Theorem with the singular values partitioned into $\{\lambda_2\}$ and $\{\lambda_i\}_{i \neq 2}$, we have $\sin\theta(\boldsymbol{u}_2, \tilde{\boldsymbol{u}}_2) \leq O(\frac{\epsilon}{\delta})$. Considering that $\delta = \Theta(1)$, we further have $\sin\theta(\boldsymbol{u}_1, \tilde{\boldsymbol{u}}_1) \leq O(\epsilon)$ and $\sin\theta(\boldsymbol{u}_2, \tilde{\boldsymbol{u}}_2) \leq O(\epsilon)$. Similar conclusions hold for $\tilde{\boldsymbol{v}}_i$'s as well.

Now, for $i = 1, 2$, let's examine the difference between $\tilde{\boldsymbol{u}}_i$ and $\boldsymbol{u}_i$:

$$\begin{aligned}
\|\tilde{\boldsymbol{u}}_i - \boldsymbol{u}_i\|^2 =& 2(1 - \cos\theta(\boldsymbol{u}_i, \tilde{\boldsymbol{u}}_i)) \\
=& 2(1 - \sqrt{1 - \sin\theta(\boldsymbol{u}_i, \tilde{\boldsymbol{u}}_i)}) \\
=& 2\frac{\sin\theta(\boldsymbol{u}_i, \tilde{\boldsymbol{u}}_i)}{1 + \sqrt{1 - \sin\theta(\boldsymbol{u}_i, \tilde{\boldsymbol{u}}_i)}} \\
=& O(\sin\theta(\boldsymbol{u}_i, \tilde{\boldsymbol{u}}_i)) \\
\leq& O(\epsilon).
\end{aligned}$$

Therefore, $\|\tilde{\boldsymbol{u}}_i - \boldsymbol{u}_i\| \leq O(\sqrt{\epsilon})$. Similarly, we can deduce that $\|\tilde{\boldsymbol{v}}_i - \boldsymbol{v}_i\| \leq O(\sqrt{\epsilon})$. By some algebraic calculations, we further have $\|\sum_{i=1}^2 \tilde{\lambda}_i \tilde{\boldsymbol{u}}_i \tilde{\boldsymbol{v}}_i^\top - \boldsymbol{G}^*\|_2 \leq O(\sqrt{\epsilon})$. Now, if $p \geq 3$, we previously established that $|\tilde{\lambda}_i| \leq \epsilon$ for $i \geq 3$. Consequently, $\|\sum_{i=3}^p \tilde{\lambda}_i \tilde{\boldsymbol{u}}_i \tilde{\boldsymbol{v}}_i^\top\|_2 = \max\{\tilde{\lambda}_i\}_{i=3}^p \leq \epsilon$. Thus, $\|\sum_{i=1}^p \tilde{\lambda}_i \tilde{\boldsymbol{u}}_i \tilde{\boldsymbol{v}}_i^\top - \boldsymbol{G}^*\|_2 \leq \|\sum_{i=1}^2 \tilde{\lambda}_i \tilde{\boldsymbol{u}}_i \tilde{\boldsymbol{v}}_i^\top - \boldsymbol{G}^*\|_2 + \|\sum_{i=3}^p \tilde{\lambda}_i \tilde{\boldsymbol{u}}_i \tilde{\boldsymbol{v}}_i^\top\|_2 \leq O(\sqrt{\epsilon})$. ■

For convenience, let's define

$$\epsilon_0 := \frac{\sigma_{\xi,T} + \sigma_{\xi,I}}{\sqrt{n}} + \frac{\sigma_{\xi,I}\sigma_{\xi,T}}{n} + \frac{\sigma_{\xi,A}\sqrt{\log n} + \sigma_{\xi,B}\sqrt{\log n} + \sigma_{\xi,I}\sigma_{\xi,T}}{n} + \sqrt{\frac{\log n}{n}}.$$

Corollary 32 *The minimizer satisfies the following with a probability of at least $1 - O(\frac{1}{\text{poly}(n)})$,*

$$\|\boldsymbol{W}_I^{*\top}\boldsymbol{W}_T^* - \frac{1}{\rho}\boldsymbol{D}_I \begin{bmatrix} 1 + \sigma_{core}^2 & 2p - 1 \\ 2p - 1 & 1 + \sigma_{spu}^2 \end{bmatrix} \boldsymbol{D}_T^\top\|_2 \leq \frac{1}{\rho} O(\sqrt{\epsilon_0}).$$

G.2.4 Zero-shot classification

In the following analysis, we will examine the zero-shot accuracy in an event where

$$\|\boldsymbol{W}_I^{*\top}\boldsymbol{W}_T^* - \frac{1}{\rho}\boldsymbol{D}_I \begin{bmatrix} 1 + \sigma_{\text{core}}^2 & 2p - 1 \\ 2p - 1 & 1 + \sigma_{\text{spu}}^2 \end{bmatrix} \boldsymbol{D}_T^\top\|_2 \leq \frac{1}{\rho} O(\sqrt{\epsilon_0}).$$

It is important to note that such an event occurs with a probability of at least $1 - O(\frac{1}{\text{poly}(n)})$ by Corollary 32.

Let $\boldsymbol{x}_I = \boldsymbol{D}_I \begin{bmatrix} y+\zeta_1 \\ a+\zeta_2 \end{bmatrix} + \boldsymbol{\xi}_I$ be a test input satisfies $y = 1, a = -1$.

By Lemma 20, with probability at least $1 - O(\frac{1}{\text{poly}(n)})$

$$\|\boldsymbol{x}_I\| \leq O(\log n). \tag{10}$$

Recall that the prompts are $\boldsymbol{p}_y = \boldsymbol{D}_T \begin{bmatrix} y \\ 0 \end{bmatrix}$ for $y = -1, 1$. Then

$$\begin{aligned} \left| \boldsymbol{x}_I^\top \boldsymbol{W}_I^{*\top} \boldsymbol{W}_T^* \boldsymbol{x}_T^{(y)} - \frac{1}{\rho} \boldsymbol{x}_I^\top \boldsymbol{D}_I \begin{bmatrix} 1+\sigma_{\text{core}}^2 & 2p-1 \\ 2p-1 & 1+\sigma_{\text{spu}}^2 \end{bmatrix} \boldsymbol{D}_T^\top \boldsymbol{x}_T^{(y)} \right| \leq & \|\boldsymbol{x}_I\| \|\boldsymbol{x}_T^{(y)}\| \frac{1}{\rho} O(\sqrt{\epsilon_0}) \\ \leq & \frac{1}{\rho} O(\sqrt{\epsilon_0} \log n). \end{aligned} \tag{11}$$

Now let's look at $\boldsymbol{x}_I^\top \boldsymbol{D}_I \begin{bmatrix} 1+\sigma_{\text{core}}^2 & 2p-1 \\ 2p-1 & 1+\sigma_{\text{spu}}^2 \end{bmatrix} \boldsymbol{D}_T^\top \boldsymbol{x}_T^{(y)}$.

$$\boldsymbol{x}_I^\top \boldsymbol{D}_I \begin{bmatrix} 1+\sigma_{\text{core}}^2 & 2p-1 \\ 2p-1 & 1+\sigma_{\text{spu}}^2 \end{bmatrix} \boldsymbol{D}_T^\top \boldsymbol{x}_T^{(y)} = y \left((1+\zeta_1)(1+\sigma_{\text{core}}^2) + (-1+\zeta_2)(2p-1) \right). \tag{12}$$

In order for the model to make correct predictions for this example, we need $\boldsymbol{x}_I^\top \boldsymbol{W}_I^{*\top} \boldsymbol{W}_T^* \boldsymbol{x}_T^{(1)} > \boldsymbol{x}_I^\top \boldsymbol{W}_I^{*\top} \boldsymbol{W}_T^* \boldsymbol{x}_T^{(-1)}$. Based on Equation 11, we can establish the following sufficient condition:

$$\begin{aligned} & \frac{1}{\rho} \boldsymbol{x}_I^\top \boldsymbol{D}_I \begin{bmatrix} 1+\sigma_{\text{core}}^2 & 2p-1 \\ 2p-1 & 1+\sigma_{\text{spu}}^2 \end{bmatrix} \boldsymbol{D}_T^\top \boldsymbol{x}_T^{(1)} - \frac{1}{\rho} O(\sqrt{\epsilon_0} \log n) \\ > & \frac{1}{\rho} \boldsymbol{x}_I^\top \boldsymbol{D}_I \begin{bmatrix} 1+\sigma_{\text{core}}^2 & 2p-1 \\ 2p-1 & 1+\sigma_{\text{spu}}^2 \end{bmatrix} \boldsymbol{D}_T^\top \boldsymbol{x}_T^{(-1)} + \frac{1}{\rho} O(\sqrt{\epsilon_0} \log n). \end{aligned}$$

By substituting Equation 12 into the above expressions, we obtain:

$$(1+\zeta_1)(1+\sigma_{\text{core}}^2) + (-1+\zeta_2)(2p-1) - O(\sqrt{\epsilon_0} \log n) > 0. \tag{13}$$

Let ϵ_1 denote last term on the LHS, i.e., $\epsilon_1 = O(\sqrt{\epsilon_0} \log n)$.

By recognizing that $\zeta_1(1+\sigma_{\text{core}}^2) + \zeta_2(2p-1)$ is a variable follows the Gaussian distribution $\mathcal{N}\left(0, (1+\sigma_{\text{core}}^2)^2 \sigma_{\text{core}}^2 + (2p-1)^2 \sigma_{\text{spu}}^2\right)$, we can derive the following probability:

$$\begin{aligned} & \Pr\left((1+\zeta_1)(1+\sigma_{\text{core}}^2) + (-1+\zeta_2)(2p-1) - \epsilon_1 > 0 \right) \\ = & \Pr_{v \sim \mathcal{N}(0,1)} \left(v > \frac{2p-2-\sigma_{\text{core}}^2 + \epsilon_1}{\sqrt{(1+\sigma_{\text{core}}^2)^2 \sigma_{\text{core}}^2 + (2p-1)^2 \sigma_{\text{spu}}^2}} \right) \\ = & 1 - \boldsymbol{\Phi}\left(\frac{2p-2-\sigma_{\text{core}}^2 + \epsilon_1}{\sqrt{(1+\sigma_{\text{core}}^2)^2 \sigma_{\text{core}}^2 + (2p-1)^2 \sigma_{\text{spu}}^2}} \right), \end{aligned} \tag{14}$$

where $\boldsymbol{\Phi}(\cdot)$ denotes the CDF of the standard normal distribution.

Therefore, we can conclude that, in order for the model to make correct predictions, the failure probability is bounded by $\boldsymbol{\Phi}\left(\frac{2p-2-\sigma_{\text{core}}^2 + \epsilon_1}{\sqrt{(1+\sigma_{\text{core}}^2)^2 \sigma_{\text{core}}^2 + (2p-1)^2 \sigma_{\text{spu}}^2}} \right)$ plus the probability for

Equation 10 to not hold. Thus, the error rate on test examples where $y = 1, a = -1$, denoted by $\mathbf{Err}_{(1,-1)}$, is bounded by:

$$\begin{aligned}
&\mathbf{Err}_{(1,-1)}(\boldsymbol{W}_I^*, \boldsymbol{W}_T^*) \\
\leq &\boldsymbol{\Phi}\left(\frac{2p - 2 - \sigma_{\text{core}}^2 + \epsilon_1}{\sqrt{(1+\sigma_{\text{core}}^2)^2\sigma_{\text{core}}^2 + (2p-1)^2\sigma_{\text{spu}}^2}}\right) + O\left(\frac{1}{\text{poly}(n)}\right) \\
= &\boldsymbol{\Phi}\left(\frac{2p - 2 - \sigma_{\text{core}}^2}{\sqrt{(1+\sigma_{\text{core}}^2)^2\sigma_{\text{core}}^2 + (2p-1)^2\sigma_{\text{spu}}^2}}\right) + O\left(\frac{\epsilon_1}{\sqrt{(1+\sigma_{\text{core}}^2)^2\sigma_{\text{core}}^2 + (2p-1)^2\sigma_{\text{spu}}^2}} + \frac{1}{\text{poly}(n)}\right) \quad ① \\
= &\boldsymbol{\Phi}\left(\frac{2p - 2 - \sigma_{\text{core}}^2}{\sqrt{(1+\sigma_{\text{core}}^2)^2\sigma_{\text{core}}^2 + (2p-1)^2\sigma_{\text{spu}}^2}}\right) + O\left(\epsilon_1 + \frac{1}{\text{poly}(n)}\right).
\end{aligned}$$

Note that Equation ① is obtained by taking the first order Taylor approximation for $\boldsymbol{\Phi}$.

We can also derive that $\mathbf{Err}_{(-1,1)}$ is bounded in the same way as above. Similarly, we can obtain

$$\mathbf{Err}_{(1,1)} = \mathbf{Err}_{(-1,-1)} \leq \boldsymbol{\Phi}\left(\frac{-2p - \sigma_{\text{core}}^2}{\sqrt{(1+\sigma_{\text{core}}^2)^2\sigma_{\text{core}}^2 + (2p-1)^2\sigma_{\text{spu}}^2}}\right) + O\left(\epsilon_1 + \frac{1}{\text{poly}(n)}\right). \quad (15)$$

Converting error rate to accuracy yields Theorem 6. Note that the error rate can be lower bounded with the same non-negligible term. For example, $\mathbf{Err}_{(-1,-1)} \geq \boldsymbol{\Phi}\left(\frac{-2p-\sigma_{\text{core}}^2}{\sqrt{(1+\sigma_{\text{core}}^2)^2\sigma_{\text{core}}^2+(2p-1)^2\sigma_{\text{spu}}^2}}\right) - o(1)$.

G.3 MMCL with Feature Masking

With feature masking, the proof is almost the same as above, but we just need to realize that the only change is in the covariance of features, thus we would get

$$\|\boldsymbol{W}_I^{*\top}\boldsymbol{W}_T^* - \frac{1}{\rho}\boldsymbol{D}_I \begin{bmatrix} 1+\pi_{\text{core}}\sigma_{\text{core}}^2 & 2p-1 \\ 2p-1 & 1+\pi_{\text{spu}}\sigma_{\text{spu}}^2 \end{bmatrix} \boldsymbol{D}_T^\top\|_2 \leq \frac{1}{\rho}O(\sqrt{\epsilon_0}).$$

Then going through the same steps as in Section G.2.4 yields Theorem 12.

Appendix H. Analysis for Data Model 2

H.1 Analysis for SL

For a vector $\boldsymbol{v}$, we use $\boldsymbol{v}[j]$ to denote its j-th element. W.O.L.G., let $\boldsymbol{D}_I \in \mathbb{d}_I \times l$ be the first l-th columns of an identity matrix (since by definition $\boldsymbol{D}_I$ has orthonormal columns and we can always apply a change of basis).

By definition, the solution of SVM, denoted by $\boldsymbol{W}^*$, satisfies

$$\boldsymbol{W}^* = \arg\min_{\boldsymbol{W}=[\boldsymbol{w}_1 \ \dots \ \boldsymbol{w}_{2m}]} \|\boldsymbol{W}\| \ s.t. \ \boldsymbol{w}_y^\top \boldsymbol{x}_I - \boldsymbol{w}_{y'}^\top \boldsymbol{x}_I \geq 1, \ \forall (\boldsymbol{x}_I, y) \text{ and } y' \neq y \text{ in the training set.} \quad (16)$$

Construct $\hat{\boldsymbol{W}} = [\hat{\boldsymbol{w}}_1 \;\; \hat{\boldsymbol{w}}_2 \;\; \dots \;\; \hat{\boldsymbol{w}}_{2m}]$, where $\hat{\boldsymbol{w}}_{2k-1+(1+c)/2}$'s k-th element is $\frac{c}{(1-\beta)(1+\alpha^2)}$, its $(k+m)$-th element is $\frac{c\alpha}{(1-\beta)(1+\alpha^2)}$ and its other elements are zero. It is easy to check that $\hat{\boldsymbol{W}}$ satisfies the condition $\hat{\boldsymbol{w}}_y^\top \boldsymbol{x}_I - \hat{\boldsymbol{w}}_{y'}^\top \boldsymbol{x}_I \geq 1$, for any $(\boldsymbol{x}_I, y)$ and $y' \neq y$ from the training set, and $\|\hat{\boldsymbol{W}}\| = \sqrt{2m} \times \frac{1}{(1-\beta)(1+\alpha^2)} \times \sqrt{1+\alpha^2}$.

Definition 33 *Define $\mathcal{S}$ as the set of classes such that $\forall (k,c) \in \mathcal{S}$ ((k,c) is the alias of y), the following holds: $\exists$ an example $(\boldsymbol{x}_I, y)$ from $\mathscr{P}^\star$, such that $\boldsymbol{x}_I$'s $(k+m)$-th element is $-\alpha c$, and the margin maximizer on training data can make the correct prediction on this example, i.e.,*

$$[\boldsymbol{x}_I]_{k+m} = -\alpha c \quad \textit{and} \quad \arg\max_j \hat{\boldsymbol{w}}_j^\top \boldsymbol{x}_I = y. \tag{17}$$

Define $\epsilon \coloneqq |\mathcal{S}|/m$ Next, we are going to show that the assumption $\epsilon > \frac{8}{(1+\alpha^2)(1-\beta)^2 - 8}$ would lead to contradiction.

Assume $|\mathcal{S}| \geq \epsilon m$. For each $(k,c) \in \mathcal{S}$, let $\boldsymbol{x}_I^+$ be the input of an example satisfying the condition in equation 17. Then $\boldsymbol{x}_I^+$'s k-th element is c, and $(k+m)$-th element is $-\alpha c$. We construct an example $\boldsymbol{x}_I^-$ whose k-th element is c, $(k+m)$-th element is αc, and the remaining elements are the opposite of the corresponding elements in $\boldsymbol{x}_I^+$.

From assumption 17, we have

$$\forall j \neq y, \boldsymbol{w}_y^{*\top} \boldsymbol{x}_I^+ - \boldsymbol{w}_j^{*\top} \boldsymbol{x}_I^+ > 0. \tag{18}$$

Note that $\boldsymbol{x}_I^-$ shows in the training set. By the condition for the SVM solution, we have

$$\forall j \neq y, \boldsymbol{w}_y^{*\top} \boldsymbol{x}_I^- - \boldsymbol{w}_j^{*\top} \boldsymbol{x}_I^- \geq 1. \tag{19}$$

Now, for any $j \neq y$, let's compute a lower bound for $\|\boldsymbol{w}_y^* - \boldsymbol{w}_j^*\|$. Any vector $\boldsymbol{w}_y^* - \boldsymbol{w}_j^*$ can be written as $a\boldsymbol{x}_I^+ + b\boldsymbol{x}_I^- + \boldsymbol{v}_\perp$, where $\boldsymbol{v}_\perp$ is a vector orthogonal to both $\boldsymbol{x}_I^+$ and $\boldsymbol{x}_I^-$. By equations 18 and 19, we have

$$\begin{cases} c_1 a + c_2 b > 0 \\ c_2 a + c_3 b \geq 1 \end{cases}, \tag{20}$$

where

$$c_1 = \|\boldsymbol{x}_I^+\|^2 = (1+\alpha^2)\left(1 + (m-1)\beta^2\right) \tag{21}$$

$$c_2 = \boldsymbol{x}_I^{+\top} \boldsymbol{x}_I^- = 1 - \alpha^2 - (m-1)\beta^2(1+\alpha^2) \tag{22}$$

$$c_3 = \|\boldsymbol{x}_I^-\|^2 = (1+\alpha^2)\left(1 + (m-1)\beta^2\right). \tag{23}$$

Remember that we want to lower bound the following quantity

$$\|\boldsymbol{w}_y^* - \boldsymbol{w}_j^*\|^2 = c_1 a^2 + c_3 b^2 + 2c_2 ab + \|\boldsymbol{v}_\perp\|^2, \tag{24}$$

given the constraints in 20. By equations 21 to 23, $(2c_2)^2 - 4c_1 c_3 < 0$. Then $c_1 a^2 + c_3 b^2 + 2c_2 ab = D$ is always an ellipse or a circle centered at the origin in the a-b coordinate system, where a larger D means a larger radius. Also, given that $c_1 > 0, c_2 < 0, c_3 > 0$, by plotting the feasible area, we can observe that D achieves its minimum when the intersection point of

lines $c_1 a + c_2 b = 0$ and $c_2 a + c_3 b = 1$ is exactly at the ellipse (or circle). Now we can solve for the minimum of $c_1 a^2 + c_3 b^2 + 2c_2 ab$:

$$c_1 a^2 + c_3 b^2 + 2c_2 ab \geq \frac{c_1}{c_1 c_3 - c_2^2}.$$

Then by equation 24 we have

$$\|\boldsymbol{w}_y^* - \boldsymbol{w}_j^*\|^2 \geq \frac{c_1}{c_1 c_3 - c_2^2}.$$

Then we get the following lower bound for $\|\boldsymbol{w}_y^*\|^2 + \|\boldsymbol{w}_j^*\|^2$

$$\begin{aligned}
\|\boldsymbol{w}_y^*\|^2 + \|\boldsymbol{w}_j^*\|^2 \geq & \frac{(\|\boldsymbol{w}_y^*\| + \|\boldsymbol{w}_j^*\|)^2}{2} \\
\geq & \frac{(\|\boldsymbol{w}_y^* - \boldsymbol{w}_j^*\|)^2}{2} \\
\geq & \frac{1}{2} \frac{c_1}{c_1 c_3 - c_2^2} \qquad (25) \\
= & \frac{1}{8} \frac{(1+\alpha^2)\left(1 + (m-1)\beta^2\right)}{\alpha^2 \left(1 + (m-1)\beta^2\right) + (m-1)\beta^2} \\
\geq & \frac{1}{8}. \qquad (26)
\end{aligned}$$

Now since there are at least ϵm different such y's, and for each y we can pick a distinct j, we get that the sum of the squared norms of the corresponding weights is at least $\epsilon m/8$. Then we introduce the following lemma.

Lemma 34 *For any $i \neq j$, the following holds for the margin maximizer $\boldsymbol{W}^*$:*

$$\|\boldsymbol{w}_i^*\|^2 + \|\hat{\boldsymbol{w}}_j\|^2 \geq \frac{1}{(1+\alpha^2)(1-\beta)^2}$$

Proof Consider any y, y' whose aliases are (k, c) and (k', c') and $y \neq y'$. Firstly, recall the condition in Equation 16 which is

$$\boldsymbol{w}_y^{*\top} \boldsymbol{x}_I - \boldsymbol{w}_{y'}^{*\top} \boldsymbol{x}_I \geq 1, \quad \forall \boldsymbol{x}_I \text{ with label } y.$$

By the condition in Equation 16, we also have

$$\boldsymbol{w}_{y'}^{*\top} \boldsymbol{x}_I' - \boldsymbol{w}_y^{*\top} \boldsymbol{x}_I' \geq 1, \quad \forall \boldsymbol{x}_I' \text{ with label } y'.$$

Note that we assume all examples occur in the training data in the theorem. We let $\boldsymbol{x}_I$ be an example whose k'-th element is β and $k' + m$-th element is $\beta\alpha$, and let $\boldsymbol{x}_I'$ be an example whose k-th element is β and $k + m$-th element is $\beta\alpha$, and other elements are the same as in $\boldsymbol{x}_I$. Note that such examples exist. Then we have

$$\begin{aligned}
\|\boldsymbol{x}_I\|^2 = & \|\boldsymbol{x}_I'\|^2 = (1+\alpha)^2(1 + (m-1))\beta^2) \\
-\boldsymbol{x}_I^\top \boldsymbol{x}_I' = & -(1+\alpha^2)\beta(2 + (m-2)\beta).
\end{aligned}$$

Similar to the steps from equations 19 to 25, we can solve that

$$\|\boldsymbol{w}_y^*\|^2 + \|\boldsymbol{w}_{y'}^*\|^2 \geq \frac{1}{\|\boldsymbol{x}_I\|^2 + \boldsymbol{x}_I^\top \boldsymbol{x}_I'} = \frac{1}{(1+\alpha^2)(1-\beta)^2}.$$

Note that this hold for any $y \neq y'$ which completes the proof. ■

Combining equation 26 and Lemma 34 yields $\|\boldsymbol{W}^*\|^2 \geq \frac{\epsilon m}{8} + (1-\epsilon)m\frac{1}{(1+\alpha^2)(1-\beta)^2}$. Then the assumption $\epsilon > \frac{8}{(1+\alpha^2)(1-\beta)^2-8}$ yields $\|\boldsymbol{W}^*\| > \|\hat{\boldsymbol{W}}\|$, which contradicts the fact that $\|\boldsymbol{W}^*\|$ is the solution which separates the data with smallest norm. Therefore $\epsilon \leq \frac{8}{(1+\alpha^2)(1-\beta)^2-8}$. Then we can bound the accuracy on $\mathscr{P}^\star$ by

$$\frac{1}{2} \times 100\% + \frac{1}{2} \times \frac{\epsilon m}{2m} = 50\% + \frac{\epsilon}{4} \leq 50\% + \frac{2}{(1+\alpha^2)(1-\beta)^2 - 8}.$$

H.2 Analysis for MMCL

Here we analyze with the feature masking defined in Definition 13. To obtain Theorem 10, we just need to set $\pi = 1$.

As shown in Section F.1, the minimizer can be expressed in terms of SVM of $\boldsymbol{S}$. We can calculate that

$$\boldsymbol{S} = C\boldsymbol{D}_I^\top \begin{bmatrix} \frac{1+\pi(m-1)\beta^2}{m}\mathbf{I}_m & \frac{\pi\alpha}{m}\mathbf{I}_m \\ \frac{\alpha}{m}\mathbf{I}_m & \frac{1+(m-1)\beta^2}{m}\pi\alpha^2\mathbf{I}_m \end{bmatrix} \boldsymbol{D}_T, \tag{27}$$

where C is some constant.

Then, at test time, the similarity between an input $\boldsymbol{x}_I = \boldsymbol{D}_I\boldsymbol{z}$ from class y and a prompt $\boldsymbol{p}_{c'} = \boldsymbol{D}_T c'\boldsymbol{e}_{k'}$ from class (k', y') denoted by Sim, is given by

$$Sim = C\boldsymbol{z}^\top \left(\frac{1+\pi(m-1)\beta^2}{m}\boldsymbol{e}_{k'} + \frac{\alpha}{m}\boldsymbol{e}_{k'+m} \right).$$

There are two cases to consider. if $(k, c) = (k', c')$

$$Sim_1 = C(\frac{1}{m} + \frac{m-1}{m}\pi\beta^2 \pm \frac{\alpha^2}{m}). \tag{28}$$

If $k = k', c \neq c'$

$$Sim_2 = -C(\frac{1}{m} - \frac{m-1}{m}\pi\beta^2 \pm \frac{\alpha^2}{m}) \tag{29}$$

If $k \neq k'$

$$Sim_3 = C(\pm\beta(\frac{1}{m} + \frac{m-1}{m}\pi\beta^2) \pm \beta\frac{\alpha^2}{m}) \tag{30}$$

To achieve 100% accuracy, we need $Sim_1 > Sim_2, Sim_1 > Sim_3$. which yields,

$$\pi(m-1) > \frac{\alpha^2 - 1}{\beta^2} \tag{31}$$

and

$$(1-\beta)\beta^2\pi(m-1) > (1+\beta)\alpha^2 - 1 + \beta. \tag{32}$$

Note that if inequality 32 holds, then inequality 31 holds as well. Therefore we only need inequality 32, which completes Theorem 14.

We note that Theorem 10 is obtained by setting $\pi = 1$.

Appendix I. Analysis for In-distribution Accuracy

I.1 Data model 1

We provide a brief explanation about the in-distribution accuracy, starting from SL.

W.L.O.G., we can let $\boldsymbol{D}_I$ be the first l columns of the identity matrix. Then the core and spurious features show in the first two elements of the input. Let $\hat{w}_{\text{core}}$ and $\hat{w}_{\text{spu}}$ denote first two elements of the SL loss minimizer $\boldsymbol{W}^* \in \mathbb{R}^{d_I}$, respectively.

In the following, for simplicity, we assume (1) $\|[\hat{w}_{\text{core}}\ \hat{w}_{\text{spu}}]\| = \Omega(1)$; (2) $\hat{w}_{\text{spu}} > 0$. However, it's worth noting that both of these assumptions can be shown to hold with high probability through more involved analysis.

As shown in Proposition 3 in (Sagawa et al., 2020), the following holds

$$\boldsymbol{\Phi}^{-1}(0.006) \geq \frac{1-(1+c)\gamma^2 - c' - \hat{w}_{\text{spu}} - \hat{w}_{\text{core}}}{\sqrt{\hat{w}_{\text{core}}^2\sigma_{\text{core}}'^2 + \hat{w}_{\text{spu}}\sigma_{\text{spu}}'^2}} \tag{33}$$

where $c < 1/2000$, $c' < 1/1000$, $\gamma = 9/10$. After some calculation we have $R \coloneqq \frac{\hat{w}_{\text{spu}}}{\hat{w}_{\text{core}}} > 1.51$. also alignment with noise is bounded.

Lemma 1 in (Sagawa et al., 2020) shows that

$$\|\boldsymbol{W}^*\|^2 \leq u^2 + s^2\sigma_{\xi,I}'^2(1+c_1)n_{min} + \frac{s^2\sigma_{\xi,I}'^2}{n^4} = O(\frac{n}{\sigma_{\xi,I}^2}), \tag{34}$$

where $u = 1.3125$, $s = \frac{2.61}{\sigma_{\xi,I}'^2}$.

Given a test example $(\boldsymbol{x}_I, y)$, where $\boldsymbol{x}_I = \boldsymbol{D}_I \begin{bmatrix} z_{\text{core}} \\ z_{\text{spu}} \end{bmatrix} + \boldsymbol{\xi}_I$, we have

$$|\boldsymbol{W}^*\boldsymbol{x}_I - [\hat{w}_{\text{core}}\ \hat{w}_{\text{spu}}] \begin{bmatrix} z_{\text{core}} \\ z_{\text{spu}} \end{bmatrix}| \leq |\boldsymbol{W}^*\boldsymbol{\xi}_I|. \tag{35}$$

By considering that $\boldsymbol{W}^*\boldsymbol{\xi}_I$ is a Gaussian variable and applying Lemma 18 , we have

$$|\boldsymbol{W}^*\boldsymbol{\xi}_I| = O(\|\boldsymbol{W}^*\|\sigma_{\xi,I}\sqrt{\frac{\log d_I}{d_I}}), \tag{36}$$

which further gives

$$|\boldsymbol{W}^*\boldsymbol{\xi}_I| = O(n\sqrt{\frac{\log d_I}{d_I}}) \tag{37}$$

by equation 34. Thus with sufficiently large d, $|\boldsymbol{W}^*\boldsymbol{\xi}_I| = o(1)$. Given that $\|[\hat{w}_{\text{core}}\ \hat{w}_{\text{spu}}]\|$ is at least constant, the prediction is dominated by $[\hat{w}_{\text{core}}\ \hat{w}_{\text{spu}}]\begin{bmatrix} z_{\text{core}} \\ z_{\text{spu}} \end{bmatrix}$.

Let's begin by considering test examples where $a = y$. By following similar steps as in equations 13 and 14, we can get the accuracy on such examples $\boldsymbol{\Phi}(\frac{1+R}{\sqrt{\sigma_{\text{core}}^2+R^2\sigma_{\text{spu}}^2}}) \pm o(1)$. In the scenario where $\sigma_{\text{core}} = 1$ and $\sigma_{\text{spu}} = 0$, given that $R > 1.51$, the above accuracy is $\boldsymbol{\Phi}(2.51) = 99.4\% \pm o(1)$. Given that in the in-distribution test data, such examples occur with probability $p = 1 - o(1)$, the overall in-distribution accuracy is $\boldsymbol{\Phi}(2.51) = 99.4\% \pm o(1)$.

For MMCL, from Section G.2.4, the accuracy on examples where $a = y$ is

$$1 - \boldsymbol{\Phi}(\frac{-2p - \sigma_{\text{core}}^2}{\sqrt{(1+\sigma_{\text{core}}^2)^2\sigma_{\text{core}}^2 + (2p-1)^2\sigma_{\text{spu}}^2}}) \pm o(1).$$

Considering that $p_{\text{spu}} = 1 - o(1)$, the in-distribution accuracy is also

$$1 - \boldsymbol{\Phi}(\frac{-2p - \sigma_{\text{core}}^2}{\sqrt{(1+\sigma_{\text{core}}^2)^2\sigma_{\text{core}}^2 + (2p-1)^2\sigma_{\text{spu}}^2}}) \pm o(1).$$

In the case where $\sigma_{\text{core}} = 1, \sigma_{\text{spu}} = 0$, the above accuracy is $1-\boldsymbol{\Phi}(-1.5)\pm o(1) = 93.93\%\pm o(1)$.

Therefore we see that SL has slightly higher in-distribution accuracy.

I.2 Data model 2

For data model 2, it is evident that SL can achieve 100% in-distribution accuracy, as under our assumption, every possible example shows in the training set and they are seperable. For MMCL, every example in $\mathscr{P}^{Tr}$ also shows in $\mathscr{P}^{\star}$. The fact that it achieves 100% accuracy on $\mathscr{P}^{\star}$, as we have already proved, implies a 100% in-distribution accuracy.

Appendix J. Experimental Details

J.1 The semi-synthetic Experiment

Image generation. Firstly, we define three types of blue: dark blue, medium blue, and light blue, as well as three types of red: dark red, medium red, and light red. Then, we modify images in the MNIST dataset. To generate the training data, for each image with a digit between 0 and 4, there is a 99.5% probability that the background will be colored with a random shade of blue from the three types; otherwise, it will be colored with a random shade of red. Similarly, for each image with a digit between 5 and 9, there is a 99.5% probability that the background will be colored with a random shade of red from the three types; otherwise, it will be colored with a random shade of blue. The task is to classify whether the digit in an image falls between 0-4 or 5-9. To generate the test data, the color is uniformly selected from all six colors for each image.

Caption generation. We simulate captions using vectors. Each caption is a 200-dimensional vector generated as $\boldsymbol{v} + \boldsymbol{\xi}$, where $\boldsymbol{\xi} \sim \mathcal{N}(0, \frac{1}{2000}\mathbf{I}_{200})$ and $\boldsymbol{v} = [a, b, 0, 0, 0, \ldots, 0]^\top$ with a, b generated as follows.

First, we define a dictionary that assigns a value to each color and digit:

```
DICT = {0: −4.5, 1: −3.5, 2: −2.5, 3: −1.5, 4: −0.5, \\
5: 0.5, 6: 1.5, 7: 2.5, 8: 3.5, 9: 4.5, \\
"dark blue": −2.5, "medium blue": −1.5, "light blue": −0.5,\\
"light red": 0.5, "medium red": 1.5, "dark red": 2.5  }
```

We also define the following values as the means for each category:

$$\text{mean}_{\text{0-4}} = \frac{1}{4}\sum_{d=1}^{4} \text{DICT}[d] = -2.5$$

$$\text{mean}_{\text{5-9}} = \frac{1}{4}\sum_{d=5}^{9} \text{DICT}[d] = 2.5$$

$$\text{mean}_{\text{blue}} = \frac{\text{DICT}[\text{“dark blue"}] + \text{DICT}[\text{“medium blue"}] + \text{DICT}[\text{“light blue"}]}{3} = -1.5$$

$$\text{mean}_{\text{red}} = \frac{\text{DICT}[\text{“dark red"}] + \text{DICT}[\text{“medium red"}] + \text{DICT}[\text{“light red"}]}{3} = 1.5$$

For an image with digit d and color c, the corresponding a and b are given by:

$$a = \begin{cases} \text{DICT}[d], & \text{with probability } \pi_{\text{core}} \\ \text{mean}_{\text{0-4}} \text{ if } d \in \{0,1,2,3,4\} \text{ else } \text{mean}_{\text{5-9}}, & \text{with probability } 1-\pi_{\text{core}} \end{cases}$$

$$b = \begin{cases} \text{DICT}[c], & \text{with probability } \pi_{\text{spu}} \\ \text{mean}_{\text{blue}} \text{ if } c \text{ is a kind of blue else } \text{mean}_{\text{red}}, & \text{with probability } 1-\pi_{\text{spu}}. \end{cases}$$

Training details. For MMCL, we choose LeNet with an output dimension of 128 as our vision encoder. For the ‘language’ model, since the captions are represented as vectors, we employ a linear model with an output dimension of 128. We use the CLIP loss, but without normalization when computing similarity. We use momentum SGD as the optimizer with a learning rate of 0.01, weight decay of 0.001, momentum of 0.9, a batch size of 128. The model is trained for 100 epochs. For SL, we train a LeNet using momentum SGD with a learning rate of 0.01, weight decay of 0.001, momentum of 0.9, a batch size of 128, for 40 epoch to minimize the Cross-Entropy loss.

In distribution accuracy. To measure the in-distribution test accuracy, we evaluate the models on a dataset constructed in the same way as the training data but with images from the MNIST test set. Figure 7 presents the results, showing that supervised learning achieves the highest in-distribution accuracy. This indicates that the improvement in out-of-distribution accuracy shown in Figure 2a can only be attributed to MMCL's superior robustness. In other words, MMCL enhances both effective robustness and relative robustness, as defined in (Taori et al., 2020).

J.2 Experiments on MS COCO

Datasets. The MSCOCO dataset Lin et al. (2014), or Microsoft Common Objects in COntext, is a comprehensive computer vision dataset comprising diverse images annotated with object masks and captions. We train our model on MSCOCO and test on 6 different variants of ImageNet Russakovsky et al. (2015a) as follows. ImageNet1K, a subset of the

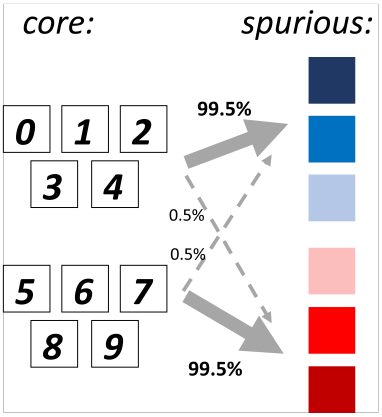


Figure 6: Construction of images

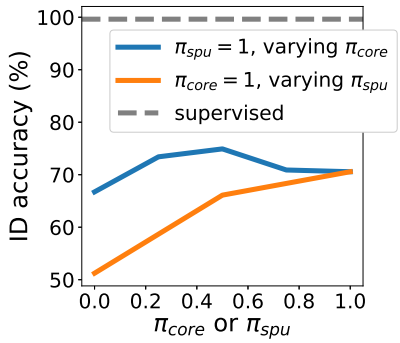


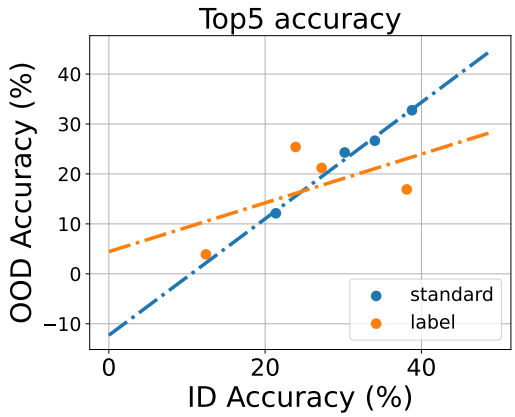
Figure 7: In-distribution test accuracy evaluated on a dataset constructed in the same way as the training data but with images from the MNIST testset.

ImageNet dataset, contains over a million high-resolution images for image classification. ImageNetv2Recht et al. (2019) addresses biases and offers a balanced category distribution. ImageNet Sketch Wang et al. (2019) features hand-drawn object sketches, while ImageNet-A Hendrycks et al. (2021b) serves as an adversarial benchmark. ImageNet-R Hendrycks et al. (2021a) offers real-world, unfiltered images, and ObjectNet Barbu et al. (2019) emphasizes robustness with objects in varied, cluttered scenes.

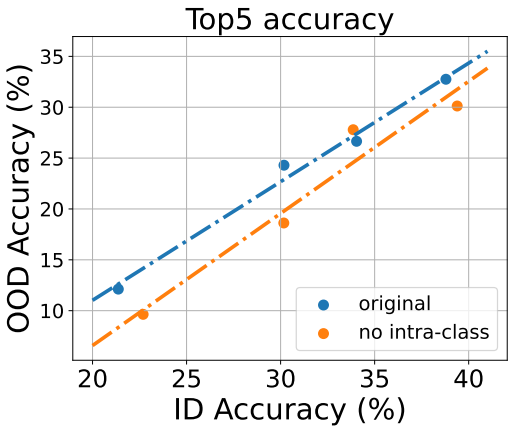
Settings. We use the official Pytorch implementation of CyCLIP Goel et al. (2022)[3]. We train CLIP (with ResNet50 as the image encoder and Transformer as the text encoder) on MSCOCO for 100 epochs with a batch size of 512. Other hyperparameters are set to their default values. Each experiment is run on 2 NVIDIA A6000.

Evaluation. Following the measurement proposed in Taori et al. (2020), which has become a standard way of measuring robustness, we plot out-of-distribution accuracy against in-distribution accuracy. The out-of-distribution accuracy is the average of the test accuracy on 6 shifted versions of ImageNet. To obtain different accuracy pairs(different points in Figures 2b abd 2c), we train models on 25%, 50%, 75%, and 100% of the original dataset

3. https://github.com/goel-shashank/CyCLIP



(a) Comparison of robustness between standard caption and label-as-caption.



(b) The relationship between ID accuracy and OOD accuracy when changing p_{drop}.

Figure 8: Additional robustness results on MSCOCO measured by top5 accuracy when changing the dataset size.

and perform the zero-shot 80-class classification task on the validation set of MSCOCO and the test set of the other datasets. Note that, for ImageNet variants, we only sample a subset of classes that can be mapped to MSCOCO classes (320 out of 1000).

Additional results. Other than the top1 accuracy, we also compute the top5 accuracy for zero-shot classification on all datasets. Figure 8a illustrates that training CLIP on the standard caption dataset yields better robustness than training on the reduced caption dataset. This validates our theoretical results in Section D about the benefits of rich image captions. As can be seen from Figure 8b, the blue line ($p_{\text{drop}} = 1$) is above the orange line ($p_{\text{drop}} = 0$). This result emphasizes that reducing intra-class contrasting does harm the robustness of MMCL as analyzed in Section C.1.

J.3 Experiments on Conceptual Captions

Datasets. Conceptual Caption (CC3M) Sharma et al. (2018) consists of around 3 million image-caption pairs which are collected and processed from Internet webpages. The dataset is divided into Training, Validation, and Test splits. The Training split includes 3,318,333 pairs in which a subset of 2,007,528 has machine-generated labels Ng et al. (2020). Utilizing the image labels in the subset of CC3M's Training split, we filter a subset of CC3M training images whose predicted labels belong to ImageNet classes with a confidence score of at least 0.6. To mitigate the effect of class imbalance, we further select classes with at least 100 but no more than 10,000 samples. The resulting subset consists of 296,801 images corresponding to 316 classes of ImageNet. To create the training and validation datasets, we split the subset with the 7:3 ratio in a stratified fashion.

Models. Due to the limited time and compute resource constraints, we consider a simplified setting similar to the one used in Ren and Li (2023). We use the pre-trained ResNet50 of CLIP, followed by a 3-layer fully connected network (MLP) with batch norm between layers for the image encoder part. During training, we freeze the ResNet50 part

and solely train the three-layer MLP. Similarly, for the text part, we employ a separate MLP with the same size on top of the pre-trained Transformer of CLIP and only train the MLP. The hidden and the output dimensions of this MLP are selected from the set $\{16, 32, 64, 128, 256, 512, 768, 1024, 2048, 4096, 8192\}$ to produce different accuracy pairs in Figures 3 and 4. Note that this setting is different from Appendix J.2 where we vary the dataset size instead.

Settings of CLIP. We use the same Pytorch implementation of CyCLIP as the settings in Appendix J.2. We train CLIP on CC3M for 100 epochs with a batch size of 1024. Other hyperparameters are set to their default values. Each experiment is run on 1 NVIDIA A6000.

Settings of SL, SupCon, and SimCLR. We train the same MLP as in CLIP on the pre-computed embeddings of images but with different losses for SL, SupCon, and SimCLR. The image embeddings are computed using the pre-trained CLIP image encoder. For a fair comparison, we also train them for 100 epochs with a batch size of 1024. Other parameters are set to the same values as the settings of CLIP.

Evaluation. We train different models on the training set and perform zero-shot 316-class classification on the validation set for CLIP. For SupCon and SimCLR, following the standard linear evaluation procedure, we discard the projection head and additionally train a linear classifier on the representations to perform classification. Similar to Appendix J.2, all models are evaluated in the test set of 6 ImageNet variants and the average accuracy on 6 datasets is considered the OOD accuracy.

Appendix K. Comparison between SupCon, Self-Supervised-CL and MMCL

K.1 A Detailed Discussion

While the original analysis in the main paper has already thoroughly demonstrated the mechanisms leading to MMCL's robustness, an in-depth comparison between SupCon, Self-Supervised-CL (e.g., SimCLR), and MMCL offers an alternative interpretation of the same findings. We hope this can further illustrate the roles of two crucial elements in MMCL: contrasting between individual examples and multimodality.

1. SupCon $<$ Self-Supervised-CL: Role of contrasting between individual examples. Let's compare the representations learned by two different unimodal representation learning techniques: SupCon and Self-Supervised-CL. Although their loss functions are quite similar, Self-Supervised-CL contrasts any two different examples, while SupCon contrasts only those with different labels. We have the following important conclusions: (1) Self-Supervised-CL closely resembles MMCL but within a single modality. Consequently, its learned representations exhibit a similar structure to MMCL's representations in that modality. This includes learning large-variance features and features shared between classes. (2) In contrast, SupCon's representations exacerbate the issue of spurious correlations because it maps both core and spurious features to the same direction in representations, making them entangled. **We theoretically demonstrate this in Theorems 35 and 36 in the next subsection**.

2. Self-Supervised-CL $<$ MMCL: Role of multi-modality. Now, one may ask, given that Self-Supervised-CL achieves good representation structures, can it achieve the same level of robustness as MMCL? Well, (1) No, because Self-Supervised-CL solely learns

unimodal (e.g., image) representations. To enable classification, we rely on supervised learning on these representations, which, as we have already shown, is not robust. (2) Yes, **only if one could** align representations of language and image modalities for zero-shot classification (bypassing supervised learning) after separately learning representations in each modality with Self-Supervised-CL. **However, this essentially falls into the category of MMCL because MMCL precisely performs these two tasks – contrasting individual examples and aligning between modalities –simultaneously!** An alternative way to think about this is that even if we adopt MMCL's image representations, training a linear classifier on the representations instead of conducting zero-shot classification would lead reduced robustness. Evidence for this can be found in Figures 14 and 15 in the CLIP paper (Radford 2021), where zero-shot classification outperforms logistic regression/few-shot learning on representations.

K.2 Theoretical Analysis for SupCon

We consider the following supervised contrastive loss, which is naturally analogous to the MMCL loss used in the main paper, and is akin to the linear loss widely adopted in theoretical CL papers in the literature Ji et al. (2021); Nakada et al. (2023).

$$\mathcal{L}_{\text{SupCon}}(\boldsymbol{W}) = -\frac{1}{n}\sum_{i}\frac{1}{N_i^+}\sum_{j \| i}\left(g(\boldsymbol{x}_{I,i})^\top g(\boldsymbol{x}_{I,j}) - \sum_{k \nparallel i}\frac{g(\boldsymbol{x}_{I,i})^\top g(\boldsymbol{x}_{I,k})}{N_i^-}\right) + \frac{\rho}{2}\|\boldsymbol{W}^\top \boldsymbol{W}\|_F^2.$$

where $j \parallel i$ represents that examples i and j are from the same class, $k \nparallel i$ represents that examples i and k are from different classes, $N_i^+ = |\{j \mid j \parallel i\}|$ and $N_i^- = |\{k \mid k \nparallel i\}|$. We consider a linear model $g(\boldsymbol{x}_I) = \boldsymbol{W}\boldsymbol{x}_I$ with $\boldsymbol{W} \in \mathbb{R}^{p\times d}$. The examples are drawn from the training distribution $\mathscr{P}^{Tr}$ within the image modality.

K.2.1 Data Model 1, SupCon

Since SupCon solely learns representations, to enable classification, we need to add a classifier l on top of the encoder g. we consider a linear classifier $l(\boldsymbol{v}) = \boldsymbol{\beta}^\top \boldsymbol{v}$ where $\boldsymbol{\beta} \in \mathbb{R}^p$. The entire model, consisting of both the encoder and the added classifier, is represented as $l(g(\boldsymbol{x}_I)) = \boldsymbol{\beta}^\top \boldsymbol{W}\boldsymbol{x}$. Its prediction $\hat{y}(\boldsymbol{x}_I)$ is given by $\text{sign}(l(g(\boldsymbol{x}_I)))$. The test accuracy on the true distribution is then denoted as $\text{Acc}_{\mathscr{P}^\star}(\boldsymbol{w}, \boldsymbol{\beta}) \coloneqq \mathbb{E}_{\boldsymbol{z}, y\in\mathscr{P}^\star, \boldsymbol{x}_I = \boldsymbol{D}_I\boldsymbol{\mu}(\boldsymbol{z}) + \boldsymbol{\xi}_I}[\mathbb{1}(\hat{y}(\boldsymbol{x}_I) = y)]$.

The following theorem demonstrates that SupCon is not robust to distribution shifts, resulting in random chance accuracy across the entire true distribution $\mathscr{P}^\star$ and making almost entirely wrong predictions on examples where $a \neq y$. To simplify the analysis we consider minimizing the population loss in noiseless setting. It's important to note that similar results in general cases can be obtained using concentration bounds similar to those in Sections G.2.2 and G.2.3.

Theorem 35 *Consider Assumption 4 and additionally consider the noiseless setting where $\sigma_{\xi,I} = 0$ and let $n \to \infty$. Let $\boldsymbol{W}^*$ be the minimizer of the SupCon loss, then no linear classifier can separate the learned representations of the two classes well. More specifically,*

$\forall \boldsymbol{\beta}$,

$$Acc_{\mathscr{P}^\star}(\boldsymbol{W}^*, \boldsymbol{\beta}) \leq 50\% + o(1).$$

Meanwhile, the model's test accuracy on examples where $a \neq y$ is $o(1)$, i.e., approaching zero.

Proof First, we define

$$\boldsymbol{S}_{\text{SupCon}} \coloneqq \frac{1}{2}\Big(\sum_{y\in\{-1,1\}} \bar{\boldsymbol{x}}^{(y)}\bar{\boldsymbol{x}}^{(y)\top} - \sum_{y\in\{-1,1\}} \bar{\boldsymbol{x}}^{(y)}\bar{\boldsymbol{x}}^{(-y)\top}\Big),$$

where $\bar{\boldsymbol{x}}^{(y)} \coloneqq \frac{1}{n/2}\sum_{\boldsymbol{x}_{I,i} \text{ from class } y} \boldsymbol{x}_{I,i}$ denotes mean of examples from class y. Based on our assumption, we can easily calculate $\boldsymbol{S}_{\text{SupCon}}$ as follows

$$\boldsymbol{S}_{\text{SupCon}} = 2\boldsymbol{D}_I \begin{bmatrix} 1 & 2p_{\text{spu}} - 1 \\ 2p_{\text{spu}} - 1 & (2p_{\text{spu}} - 1)^2 \end{bmatrix} \boldsymbol{D}_I^\top.$$

Let $\sum_{i=1}^d \lambda_i \boldsymbol{u}_i \boldsymbol{u} i^\top$ represent the eigen decomposition of $\boldsymbol{S}_{\text{SupCon}}$. Similar to the explanation provided in Section F.1, by rewriting $\mathcal{L}_{\text{SupCon}}$ as a matrix factorization objective and applying the Eckart-Young-Mirsky theorem, we obtain the minimizer of the loss as follows:

$$\boldsymbol{W}^{*\top}\boldsymbol{W}^* = \frac{1}{\rho}\sum_{i=1}^{p} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^\top,$$

We can easily compute the eigen vectors and eigen values for $\boldsymbol{S}_{\text{SupCon}}$: $\lambda_1 = (2p_{\text{spu}} - 1)^2 + 1$, $\boldsymbol{\mu}_1 = \boldsymbol{D}_I \begin{bmatrix} \frac{1}{\sqrt{(2p_{\text{spu}}-1)^2+1}} \\ \frac{2p_{\text{spu}}-1}{\sqrt{(2p_{\text{spu}}-1)^2+1}} \end{bmatrix}$, and $\lambda_2 = \cdots = \lambda_d = 0$. Therefore,

$$\boldsymbol{W}^* = \sqrt{(2p_{\text{spu}} - 1)^2 + 1}\ \boldsymbol{P}[\frac{1}{\sqrt{(2p_{\text{spu}} - 1)^2 + 1}} \quad \frac{2p_{\text{spu}} - 1}{\sqrt{(2p_{\text{spu}} - 1)^2 + 1}}]\boldsymbol{D}_I^\top,$$

where $\boldsymbol{P}$ can be any $p \times 1$ vector with norm 1. Consequently

$$\begin{aligned}\boldsymbol{W}^*\boldsymbol{x} =& \sqrt{\frac{2((2p_{\text{spu}} - 1)^2 + 1)}{\rho}}\boldsymbol{P}[\frac{1}{\sqrt{(2p_{\text{spu}} - 1)^2 + 1}} \quad \frac{2p_{\text{spu}} - 1}{\sqrt{(2p_{\text{spu}} - 1)^2 + 1}}]\boldsymbol{D}_I^\top\boldsymbol{x} \\ =& \sqrt{\frac{2((2p_{\text{spu}} - 1)^2 + 1)}{\rho}}\boldsymbol{P}[\frac{1}{\sqrt{(2p_{\text{spu}} - 1)^2 + 1}} \quad \frac{2p_{\text{spu}} - 1}{\sqrt{(2p_{\text{spu}} - 1)^2 + 1}}]\boldsymbol{z}. \end{aligned} \tag{38}$$

For any $\boldsymbol{\beta}$, the test accuracy is given by

$$\text{Acc}_{\mathscr{P}^\star}(\boldsymbol{W}^*, \boldsymbol{\beta}) = \Pr(y\boldsymbol{\beta}^\top\boldsymbol{W}^*\boldsymbol{x} > 0)$$

By equation 38 and some calculation we get

$$\text{Acc}_{\mathscr{P}^\star}(\boldsymbol{W}^*, \boldsymbol{\beta}) = \Pr(\nu_1 + \nu_2 > -1 - (2p_{\text{spu}} - 1)ay)$$

where

$$\begin{aligned}\nu_1 =& y\xi_{\text{core}}\\ \nu_2 =& y(2p_{\text{spu}}-1)\xi_{\text{spu}}\end{aligned}$$

and each can be considered as a Gaussian random variable with zero mean, independent from each other. Therefore

$$\Pr(\nu_1+\nu_2 > -1-(2p_{\text{spu}}-1)ay) = \frac{1}{2}\Pr(\nu_1+\nu_2 > -2p_{\text{spu}}) + \frac{1}{2}\Pr(\nu_1+\nu_2 > 2(p_{\text{spu}}-1)).$$

Considering $\frac{\nu_1+\nu_2}{\sqrt{\sigma_{\text{core}}^2+(2p_{\text{spu}}-1)^2\sigma_{\text{spu}}^2}} \sim \mathcal{N}(0,1)$, we have

$$\begin{aligned}\text{Acc}_{\mathscr{P}^\star}(\boldsymbol{W}^*,\boldsymbol{\beta}) \leq& \frac{1}{2}\boldsymbol{\Phi}\left(\frac{2p_{\text{spu}}}{\sqrt{\sigma_{\text{core}}^2+(2p_{\text{spu}}-1)^2\sigma_{\text{spu}}^2}}\right) + \frac{1}{2}\boldsymbol{\Phi}\left(\frac{2(1-p_{\text{spu}})}{\sqrt{\sigma_{\text{core}}^2+(2p_{\text{spu}}-1)^2\sigma_{\text{spu}}^2}}\right)\\ =& \frac{1}{2}\boldsymbol{\Phi}\left(\frac{2p_{\text{spu}}}{\sqrt{\sigma_{\text{core}}^2+(2p_{\text{spu}}-1)^2\sigma_{\text{spu}}^2}}\right) + o(1) \qquad \text{because } p_{\text{spu}} = 1-o(1)\\ \leq& 50\% + o(1).\end{aligned}$$

Additionally, the accuracy on examples where $a \neq y$ is given by $\frac{1}{2}\boldsymbol{\Phi}\left(\frac{2(1-p_{\text{spu}})}{\sqrt{\sigma_{\text{core}}^2+(2p_{\text{spu}}-1)^2\sigma_{\text{spu}}^2}}\right) = o(1)$, i.e., almost zero. ■

K.3 Data Model 2, SupCon

Theorem 36 *Under the same assumption as in Theorems 9 and 10. Let $\boldsymbol{W}^*$ be the minimizer of the $\mathcal{L}_{SupCon}$ loss. If we train a multi-class linear classifier $g(\boldsymbol{f}) = \boldsymbol{B}\boldsymbol{f}$ with with $\boldsymbol{B}^{2m\times p}$ using Cross-Entropy loss on the learned representations (i.e., given by $\boldsymbol{W}^*\boldsymbol{x}_I$), the test accuracy over the true distribution is 50%. Formally, let $\boldsymbol{B}^*$ be the trained linear classifier $Acc_{\mathscr{P}^*}(\boldsymbol{W}^*,\boldsymbol{B}^*) = 50\%$. Moreover, no linear classifier can achieve accuracy better than 75%, i.e., $\forall \boldsymbol{B}$, $Acc_{\mathscr{P}^*}(\boldsymbol{W}^*,\boldsymbol{B}) \leq 75\%$.*

Proof First, we define

$$\boldsymbol{S}_{\text{SupCon}} = \frac{1}{2m-1}\sum_{y=1}^{2m}\bar{\boldsymbol{x}_I}^{(y)}\bar{\boldsymbol{x}_I}^{(y)\top},$$

where $\bar{\boldsymbol{x}}_I^{(y)} \coloneqq \frac{1}{\#\text{ examples from class } y}\sum_{\boldsymbol{x}_I \text{ from class } y}\boldsymbol{x}_I = [c\boldsymbol{e}_k^\top \quad \alpha c\boldsymbol{e}_k^\top]^\top$ for any y with alias (k,c). Here $\boldsymbol{e}_k$ denotes the k-th standard basis in $\mathbb{R}^m$. Similar to the previous subsection, by rewriting the objective and applying the Eckart-Young-Mirsky theorem, we obtain

$$\boldsymbol{W}^{*\top}\boldsymbol{W}^* = \frac{1}{\rho}\sum_{i=1}^{p}\lambda_i\boldsymbol{u}_i\boldsymbol{u}_i^\top,$$

where $\frac{1}{\rho}\sum_{i=1}^{d}\lambda_i \boldsymbol{u}_i\boldsymbol{u}_i^\top$ is the eigen decomposition of $\boldsymbol{S}_{\text{SupCon}}$.The eigenvalues and vectors can be calculated as follows:

$$\lambda_i = \frac{2}{2m-1}(1+\alpha^2), \quad \boldsymbol{u}_i = \boldsymbol{D}_I[\frac{1}{\sqrt{1+\alpha^2}}\boldsymbol{e}_i^\top \quad \frac{\alpha}{\sqrt{1+\alpha^2}}\boldsymbol{e}_i^\top]^\top, \quad \forall i \in [m]$$
$$\lambda_i = 0, \quad \forall i > m.$$

Therefore

$$\boldsymbol{W}^* = \sqrt{\frac{2(1+\alpha^2)}{2m-1}}\boldsymbol{P}\begin{bmatrix}\boldsymbol{u}_1^\top \\ \vdots \\ \boldsymbol{u}_m^\top\end{bmatrix},$$

where $\boldsymbol{P} \in \mathbb{R}^{p\times m}$ can be any matrix with orthonormal columns. Consequently, for any example $\boldsymbol{x}_I$ in class (k, c),

$$\boldsymbol{W}^*\boldsymbol{x}_I = \sqrt{\frac{2(1+\alpha^2)}{2m-1}}(\frac{c}{\sqrt{1+\alpha^2}} + \text{sign}(\boldsymbol{z}[k+m])\frac{\alpha^2}{\sqrt{1+\alpha^2}})\boldsymbol{P}\boldsymbol{e}_k. \tag{39}$$

On the training distribution, $\text{sign}(\boldsymbol{z}[k+m])$ is always c. Therefore, the linear classifier trained on the representations of the training data would be (recall the relation between CE loss and SVD in Section F.1):

$$\boldsymbol{B}^* = \begin{bmatrix}\sqrt{\frac{2m-1}{2}}\frac{1}{1+\alpha^2}\boldsymbol{e}_1^\top \\ -\sqrt{\frac{2m-1}{2}}\frac{1}{1+\alpha^2}\boldsymbol{e}_1^\top \\ \sqrt{\frac{2m-1}{2}}\frac{1}{1+\alpha^2}\boldsymbol{e}_2^\top \\ -\sqrt{\frac{2m-1}{2}}\frac{1}{1+\alpha^2}\boldsymbol{e}_2^\top\end{bmatrix}\boldsymbol{P}^\top.$$

On the true distribution, $\boldsymbol{B}^*$ can make correct predictions for examples that also show during training. However, for any example $\boldsymbol{x}_I$ from class y with alias (k, c), if its $\boldsymbol{z}$ satisfy that $\boldsymbol{z}[k+m] = -c\alpha$, it can be observed that $(\boldsymbol{B}^*\boldsymbol{W}^*\boldsymbol{x}_I)[y-c] > (\boldsymbol{B}^*\boldsymbol{W}^*\boldsymbol{x}_I)[y]$, leading to incorrect predictions. Therefore, the overall accuracy is 50%.

Next, we analyze the case for arbitrary $\boldsymbol{B}$. For any two classes $(k, -1)$ and $(k, 1)$ in the true distribution, we can group them into four groups denoted by $G_{c,\text{sign}(\boldsymbol{z}[k+m])}$, i.e., based on the combinations of c and $\text{sign}(\boldsymbol{z}[k+m])$. Since $\alpha > 1$, we have

$$\frac{-1}{\sqrt{1+\alpha^2}} - \frac{\alpha^2}{\sqrt{1+\alpha^2}} < \frac{1}{\sqrt{1+\alpha^2}} - \frac{\alpha^2}{\sqrt{1+\alpha^2}} < \frac{-1}{\sqrt{1+\alpha^2}} + \frac{\alpha^2}{\sqrt{1+\alpha^2}} < \frac{1}{\sqrt{1+\alpha^2}} + \frac{\alpha^2}{\sqrt{1+\alpha^2}}.$$

Combining this with equation 39, we conclude that the four groups lie on the same line. Moreover, going from $G_{-1,-}$ to $G_{1,1}$, we pass through $G_{1,-1}$ first and then $G_{-1,1}$. Given this order, no linear model can classify more than three out of the four groups correctly. Since this is true for any pair of classes $(k, -1)$ and $(k, 1)$, it follows that no model can perform better than 75% accuracy on the entire distribution. ■