Hidden Learning Dynamics of Capability before Behavior in Diffusion Models

author names withheld

Under Review for the Workshop on High-dimensional Learning Dynamics, 2024

Abstract

Understanding how multimodal models generalize out of distribution is a fundamental challenge in machine learning. Compositional generalization explains this by assuming the model learns concepts and how to compose them. In this work, we train diffusion models on a compositional task from synthetic data of objects of different size and colors. We introduce a concept space as a framework to understand the learning dynamics of compositional generalization. In this framework, we identify *concept signal* as a driver of compositional generalization. Next, we find that diffusion models can acquire the *capability* to compositionally generalize long before it elicits this *behavior*. Additionally, we find that the time of capability learning can be pinpointed from the concept space learning dynamics. Finally, we suggest a *embedding disentanglement* as another metric to probe the capability of a model. Overall, we make a step in understanding the emergence of compositional capabilities in diffusion models.

1. Introduction

Modern generative models are considered to have out of distribution generalization abilities, as highlighted by text-to-image models generating "avocado chair" or "astronaut on a horse"[14, 16, 17, 21, 22]. Understanding how these multimodal models generalize out of distribution is a fundamental challenge in machine learning. Compositionality is believed to be at the core of this ability, where a model learns individual concepts and how to compose them to generate novel instances, naturally suggesting the exponential increase of generalization classes with the number of concepts learned [12].

While the ability to compositionally generalize (CG) is observed to emerge with scale[13, 21], the underlying mechanisms driving this emergence is yet unknown both in model size and compute (training time). Moreover, in generative diffusion models[5, 8, 18, 20], the loss function is usually not a good indicative of its generative abilities which makes tracking the performance of diffusion models a significant challenge.

In this work, we focus on learning dynamics of diffusion models[6], which has not been addressed in depth due to the practical aspect and scale of the problems diffusion models are usually trained on. We take a step in understanding how CG emerges in diffusion models by using a synthetic dataset consisting of objects with different size and color, allowing efficient training while being enough to show compositional generalization. We identify *concept signal*, a property of the data distribution, as a main driver of compositional generalization in diffusion models and run a suite of experiments varying this signal. Our contributions are:

1. We suggest a concept space analysis of learning dynamics of diffusion models.
2. We determine that a stronger *concept signal* results in a more efficient learning of a concept.
3. We find that the capability of CG emerges long before elicitation of this behavior.
4. Based on the above findings, we highlight the practicality of the concept space.
5. We suggest *embedding disentanglement* as a metric for tracking compositional generalization capabilities.

2. Methodology

Synthetic Data We use a dataset consisting of images of circles of different size and colors at different locations. We deal with 4 classes spanned by a combination of 2 `size` and `color`: `size`$\in$ [`big=0`, `small=1`] and `color`$\in$ [`red=0`, `blue=1`]. This synthetic data was introduced in [12]. In this work we always use (`00`, `01`, `10`) as the training set and evaluate whether the model can generalize to generate `11`, small blue circles. We define the *concept signal* of a concept as the mean square error between images when this concept is modified. In this study, we change the color separation between red and blue Δ_{Color} to tune the level of *concept signal*. Please see App. A for further details.

Variational Diffusion Models We use variational diffusion models[8] as our generative model. We use the U-Net[15] architecture with conditional embeddings of the class conditioning added into each hidden representation. Please see App. A for more details.

3. Results

3.1. Concept Space Learning Dynamics (Fig. 1)

First, we suggest the concept space analysis of learning dynamics of diffusion models. We define the concept space of a diffusion model generation as the average probability predicted *for each concept* for a set of generated images from the same class. Thus our concept space is 2 dimensional in this case. We use a separately trained classifier to construct this concept space.

Fig. 1 shows the concept space trajectory of each class as the diffusion model trains. As expected, the training classes(in dashed lines) directly converges to their expected concept space representations. However, we see that the CG test class, `11` initially follows the concept space trajectory of `01`. We denote this phenomena as *concept memorization* where the model memorizes “small=red” instead of factorizing the two concepts of `color` and `size`. Depending on the *concept signal* level, the concept space trajectory leaves this memorization phase sooner or later. We find that settings where this disentanglement of concepts happened earlier also end up with a better final(at 15,000 gradient steps) concept space representations. **Thus, we conclude that *concept signal* enhances the end of *concept memorization* and drives compositional generalization.**

3.2. Capability vs. Behavior (Fig. 2, 3)

From Sec. 3.1, we identify that for low *concept signal* levels, CG is delayed since the debiasing of the concepts happen far from the generalization target. In Fig. 2, the same experiments are plotted in accuracies, which is simply the joint accuracy of `color` and `size`. We find that low

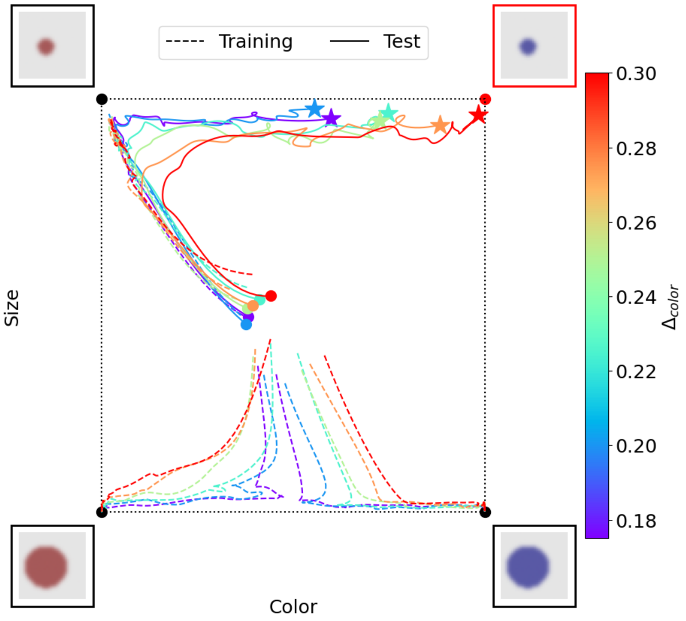

Figure 1: **Concept Space Learning Dynamics** Each trajectory corresponds to a concept space representation of a diffusion model generation at different steps during training. Dashed lines indicate training data and solid lines indicate the compositional generalization test data. Each curve is color coded by the color separation(the *concept signal*) of the data distribution.

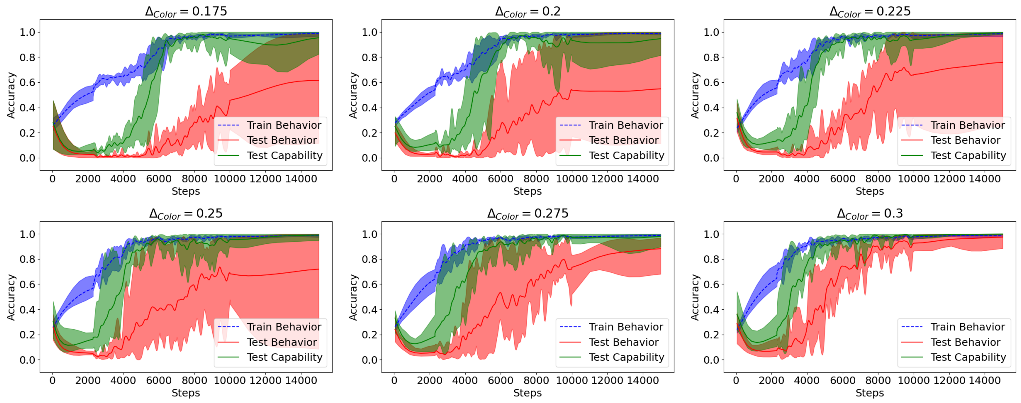

Figure 2: **Capability vs. Behavior** Each panel represents a different *concept signal* level. In each panel the test *behavior* is shown in red and the *capability* is shown in green.

signal levels not only delay CG but make its behavior seed dependent as we can see in the red area of Fig. 2. However, we hypothesize that at the moment of debiasing, where the concept space trajectory leaves the *concept memorization* phase, the *capability* of composing `color` and `size` are already present, yet the model is simply not behaving. To verify this hypothesis, we augment the conditioning prompt of the model to probe the true capability of the model (See App. B). The capability is shown in green in Fig. 2. It is evident that the CG capability is present long before the *behavior*, and moreover, the capability is learned robustly independent of the seed. **Thus, we conclude that a model's *Capability* can be present long before its *Behavior* is elicit.**

Fig. 3, shows that for this dataset, the time a capability is learned and the behavior is elicit has a simple linear relation. This suggests that the emergent *behavior* of a model can, at least in some settings, be predicted from its capabilities. Please refer to App. C for additional visualizations of the data.

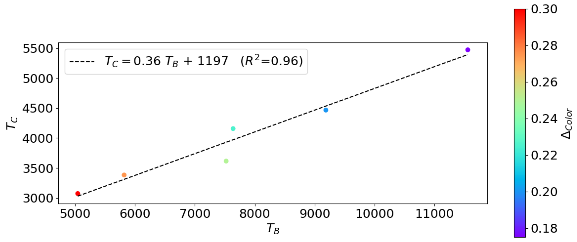

Figure 3: **Capability Learning Time vs Behavior Elicitation Time** We quantify the learning/elicitation time as when the accuracy hits 50%. The linear fit to the capability learning time and the behavior elicitation time is shown.

3.3. Practicality of Concept Space (Fig. 4)

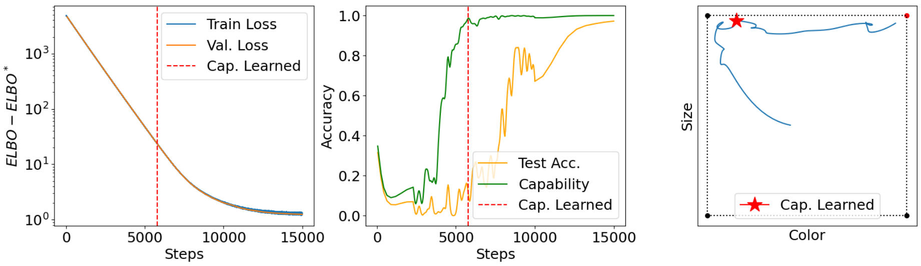

Figure 4: **Identifying Capability Learning** (Left:) The time of capability learning is invisible from the loss curves. (Center:) The test accuracy also does not clearly represent the capability learning time (Right:) The concept space gives a hint of capability learning by the clear transition in the dynamics.

Fig. 4 highlights one practical aspect of having access to the concept space learning dynamics. As one can see in the left panel, the variational lower bound(the loss function) does not show *any* hint of the time where the capability is learned. Even plotting test accuracy as in the middle panel does not directly allow one to see the existence of the capability. (We note that the capability itself is more intensive to compute (See App. B).) However, one can identify when this capability would have emerged simply by tracking the end of *concept memorization* in the concept space representation. **Thus, we conclude that visualizing the concept space can reveal hidden learning dynamics of capabilities.**

3.4. Embedding Similarity as a probe of Generalization.

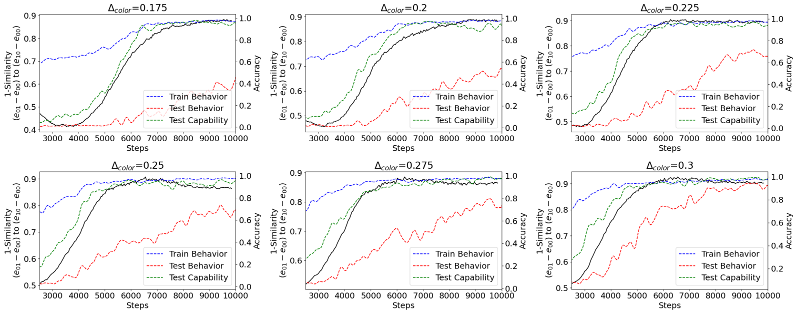

Figure 5: **Embedding Similarity of** $e_{01} - e_{00}$ **and** $e_{10} - e_{00}$

Although the concept space of generations can be simply constructed by training concept classifiers [1, 2, 4, 7, 9, 10, 19], some hypothetical concepts might be ill defined by a classifier, or perhaps not even identified by the human at the first place. It is thus important to establish methods to automatically identify the acquisition of capabilities. Here, we suggest one method restricted to our setup. Fig. 5, shows $1 - |s_{e_{01}-e_{00},e_{10}-e_{00}}|$ as a black curve, where $s_{a,b}$ is the cosine similarity of the vector a and b and e_{xx} is the conditioning embedding of the class xx. In other words, we are plotting how disentangled the size embedding ($e_{01} - e_{00}$) and the color embedding ($e_{10} - e_{00}$) is. We find that this *embedding disentanglement* is closely related to the timing a capability is learned. Please see App. D for additional probes we explored. **Thus, we conclude that model embedding based metrics for capability acquisition can be established without full generation.**

4. Conclusion

We have studied diffusion models on a synthetic data where we can analyze the learning dynamics of compositional generalization varying the *concept signal*. We discovered that 1) *concept signal* drives the onset and speed of compositional generalization. 2) Diffusion models can acquire capabilities long before it elicits a behavior. 3) Concept space analysis can help identify the acquisition of these capabilities. 4) We suggest *embedding disentanglement* as a metric to track these capabilities.

References

[1] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

[2] Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018.

[3] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2023.

[4] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *In Proc. Int. Conf. on Learning Representations (ICLR)*, 2017.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[6] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models, 2024.

[7] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018.

[8] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

[9] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*, 2017.

[10] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proc. int. conf. on machine learning (ICML)*, 2019.

[11] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[12] Maya Okawa, Ekdeep Singh Lubana, Robert P. Dick, and Hidenori Tanaka. Compositional abilities emerge multiplicatively: Exploring diffusion models on a synthetic task, 2024.

[13] Linlu Qiu, Peter Shaw, Panupong Pasupat, Tianze Shi, Jonathan Herzig, Emily Pitler, Fei Sha, and Kristina Toutanova. Evaluating the impact of model scale for compositional generalization in semantic parsing, 2022.

[14] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021.

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[16] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022.

[17] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[18] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[19] Sjoerd Van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. Are disentangled representations helpful for abstract visual reasoning? *Adv. in Neural Information Processing Systems (NeurIPS)*, 2019.

[20] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications, 2024.

[21] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022.

[22] Chenshuang Zhang, Chaoning Zhang, Mengchun Zhang, and In So Kweon. Text-to-image diffusion models in generative ai: A survey, 2023.

Appendix A. Data&Model Details

Data Details 2048 $3 \times 32 \times 32$ images per class are generated from a custom python script. Fig. 6 shows two example data distributions where one has a relatively low color separation and one has a relatively high color separation. The difference is visually mild, yet it is important in determining the diffusion model's generalization dynamics.

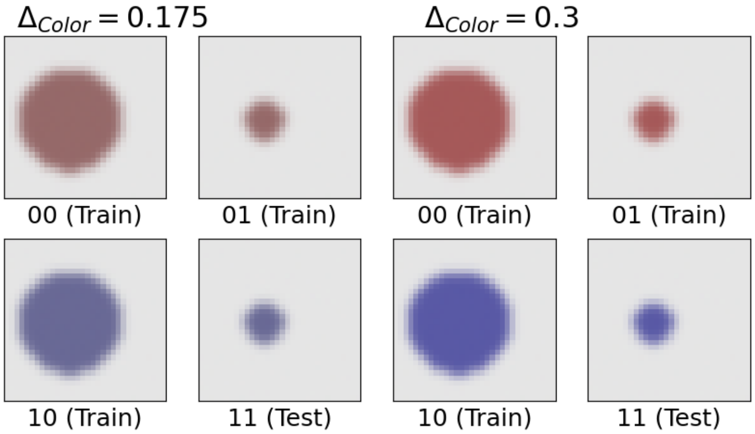

Figure 6: **Two Example Data distributions.**
Left: A data distribution with a relatively small $\Delta_{Color} = 0.175$.
Right: A data distribution with a relatively big $\Delta_{Color} = 0.3$.
These two values corresponds to the two extremes of the color separations in Fig. 1. The labels (`00`, `01`, `10`, `11`) represent (`color`, `size`).

Model Details We train a variational diffusion model[8] with a custom designed U-Net[15] architecture with [32,64,128,256] channels in each resolution block. The conditioning vectors are embedded with a 2 layer MLP with 64 hidden dimensions. GELU[3] activations are used everywhere. We train the model for 15,000 gradient steps with a AdamW[11] optimizer with learning rate 0.001 and weight decay 0.01.

Appendix B. Capability

We measure the capability of a model to generate the `11` class, small blue circles by augmenting the "prompt". Instead of using the fiducial blue conditioning of e.g. [0.4,0.4,0.6]. (The actual values depends on Δ_{Color}). We use a set of conditionings [0.0,0.0,1.0],[0.1,0.1,0.9], [0.2,0.2,0.8],[0.3,0.3,0.7],[0.4,0.4,0.6] to prompt the model to generate small blue circles. More formally, we would like to define the *Capability* of a model as the maximum accuracy over all possible prompts. The method we used is thus simply a practical implementation of this definition. We note that this definition does not make the task trivial, as there can easily be no prompt generating the right class, e.g. over-prompting for blue will generate big blue circles. In fact, this is precisely the case before capability learning happens.

Appendix C. Additional Figures for Sec. 3.2

Fig. 7 shows the capability and behavior on the compositional test class. Different Δ_{Color} are shown together to illustrate the clear effect of Δ_{Color} on capability/behavior learning times.

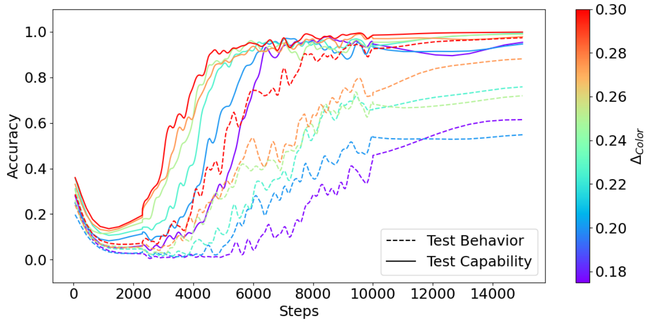

Figure 7: **Capability vs. Behavior for different** Δ_{Color} We show the learning curves of capability and behavior for different Δ_{Color}.

In Fig. 8, we show the standard deviation from different seeds in capability and behavior. We find that for low Δ_{Color} the standard deviation remains high until $15,000$ gradient steps.

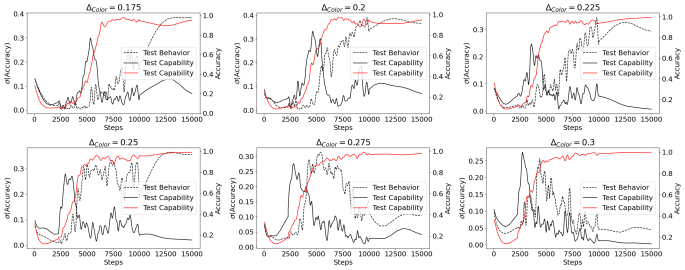

Figure 8: **Capability vs Behavior Standard Deviations** We show the standard deviation from different seeds in capability and behavior.

In Fig. 9 we show the capability and behavior in 3D trajectory plot. We see that all seeds acquire the capability while some never elicit the behavior.

Appendix D. Additional Figures for Sec. 3.4

Fig. 10, 11, 12, 13, 14, 15, 16 show additional metrics one might consider as a probe for CG. Many of these probes reflect different aspects of the train behavior, test behavior and test capability while the most clear probe is the one in Sec. 3.4.

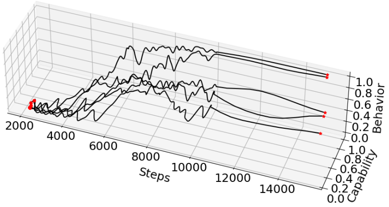

Figure 9: **3D Capability vs. Behavior** We show, for $\Delta_{Color} = 0.25$ the capability and behavior curves for different seeds. All seeds acquire the capability while some never elicit the behavior

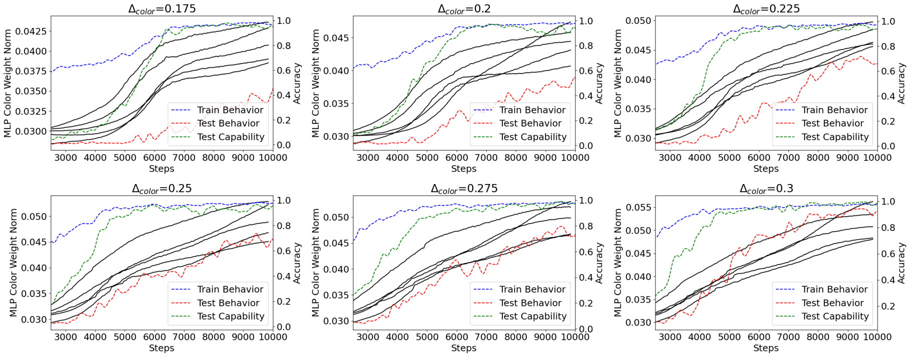

Figure 10: **MLP Weight Norms for Color**

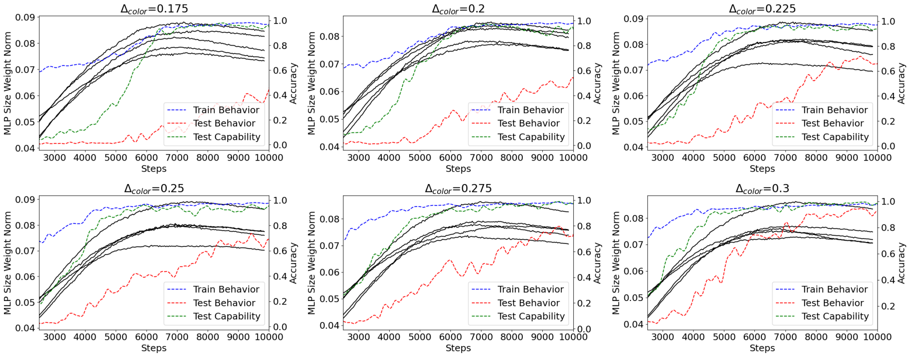

Figure 11: **MLP Weight Norms for Size**

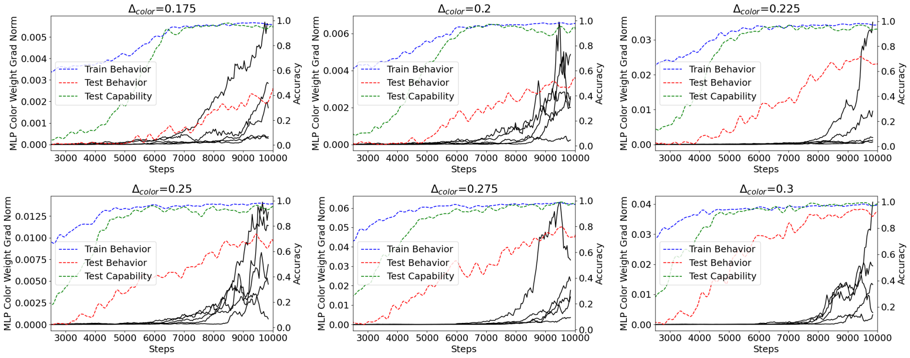

Figure 12: **MLP Gradient Norms for Color**

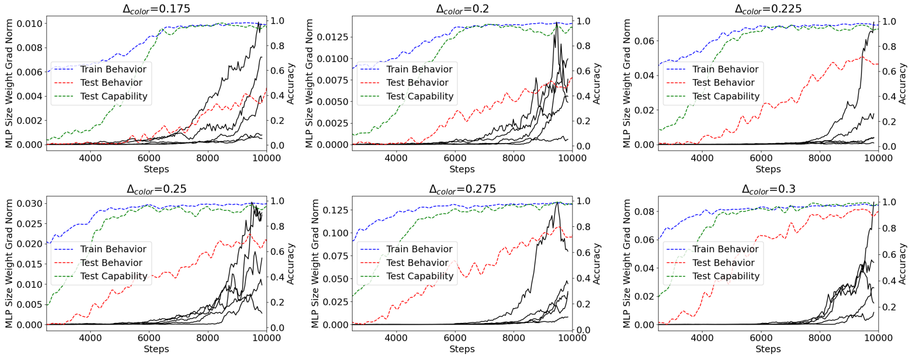

Figure 13: **MLP Gradient Norms for Size**

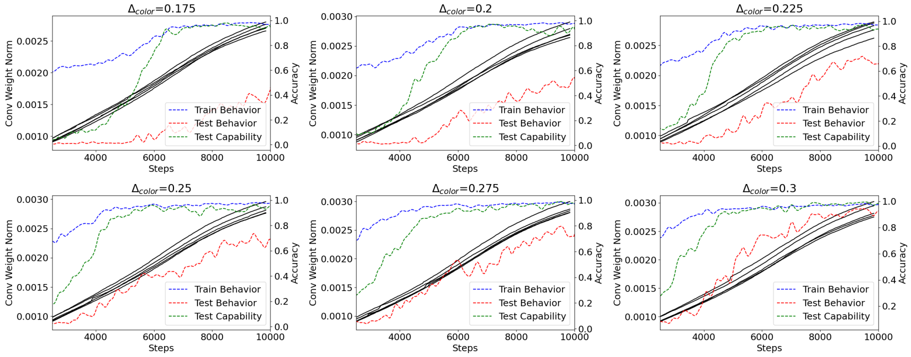

Figure 14: **Convolution Weight Norms**

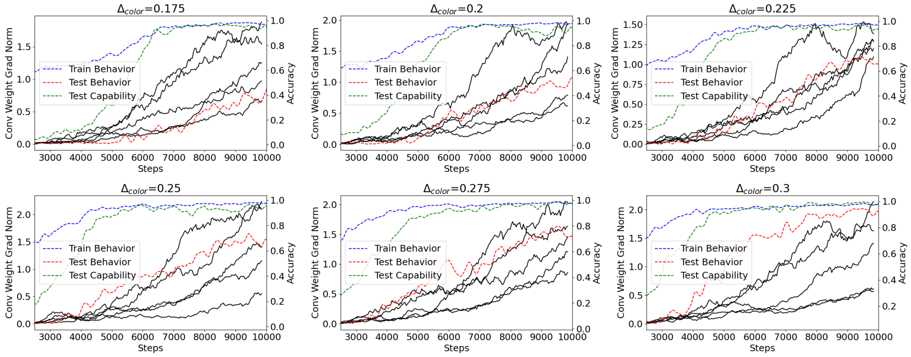

Figure 15: **Convolution Gradient Norms**

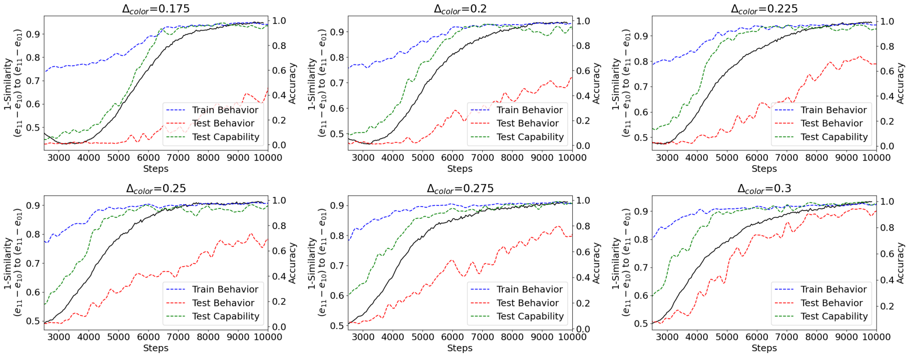

Figure 16: **Embedding Similarity of** $e_{11} - e_{10}$ **and** $e_{11} - e_{01}$