We thank all the reviewers for their insightful comments and suggestions. We will add citations and discussions of all the suggested related works in the full version. Reviewer-specific comments follow.

Reviewer 1 We consider our mutual information framework to be a core contribution of our paper. In particular, our formalization of "how well a linear classifier explains the performance of a model" has many advantages over prior formalizations (e.g. see our response to Reviewer 3). Regarding our theory example (Section 5): We separate the question of why SGD learns simple concepts first (Claim 1) from the question of why it does not forget them (Claim 2). Our current theory is only relevant to the second question, and it shows that in a simplified setting: SGD does not "forget" the simple component even when trained to completion, provided it somehow learns the simple component first. We argue that this simple example captures many properties of real settings (overparametrization, existence of non-generalizable ERMs) and hence is valuable as a step towards more general theory. Regarding Claim 1, it is true that we currently have no theoretical understanding of why linear learning occurs. We consider this one of the most important open questions of our paper, and we are attempting to make progress on this in ongoing work.

Reviewer 2 Thank you for pointing out the relevant papers. We also agree that it is scientifically valuable to describe settings where these phenomenon fail to occur (bad initialization, bad architecture/parameterization, bad optimizer, or pathological distributions). We plan on including a separate section with such examples in the final version.

Reviewer 3 Regarding novelty of our claims: Although the idea that SGD learns functions of increasing complexity has been informally floating in the community, our formalization has many advantages over prior formalizations, as described below. Notably, our metrics respect the data distribution, are independent of network-parameterization, are tractable to estimate in high dimensions, and are experimentally demonstrated for real-world distributions.

Concretely, regarding "On the spectral bias of neural networks" [1]: They consider measuring "simplicity" via the Fourier spectrum of the learnt functions. However, the Fourier decomposition is taken with respect to a *uniform* distribution on inputs. This notion is not as meaningful – if the data is not uniform (for example, natural images are certainly not uniform in pixel-space), then a function which is highly correlated with a linear function when restricted to the data distribution may appear highly non-linear with respect to the uniform distribution. And vice-versa – a function which is nearly linear *under the uniform distribution* may in fact be highly non-linear when restricted to the data distribution. Our metrics do not suffer from this issue – they are taken with respect to the true data distribution.

The synthetic experiments in [1] are all for 1-dimensional inputs, since to quote [1]: "explicitly evaluating the Fourier coefficients ... becomes prohibitively expensive for larger d (e.g. on MNIST)". Instead, their MNIST experiments are only heuristically related to the metrics they propose. In contrast, the metrics in our paper are tractable even for high-dimensional inputs, and we estimate them to high-precision on real datasets (MNIST and CIFAR).

Regarding "Understanding ... deep learning by Fourier analysis" [2]: This work also performs Fourier analysis with respect to the uniform distribution on inputs, and so suffers from the same issues as [1]. Moreover [2] *requires* that the input distribution is itself uniform to carry through the analysis[1]. That is, the theorems of [2] do not hold for non-uniform input distributions, such as images. The experiments in [2] are not relevant to our claims, since they conflate the Fourier transform in the spatial domain (i.e. 2D Fourier transforming the input image, treated as a function $\mathbb{R}^2 \to \mathbb{R}$) with the Fourier transform in function space. (i.e. Fourier transforming the classification function $\mathbb{R}^d \to \mathbb{R}$). Finally, the work of [3] is largely unrelated to our work. The authors of [3] study how the internal layers of a network vary with the *depth* of the layer, while we study how the end-to-end classification function evolves as a function of SGD steps.

Reviewer 3 brings up two concerns with the performance correlation metric. First: why do we use $\mu_Y(F; L)$ instead of simply $I(F; L)$? While it is true that $\mu_Y(F; L) \leq I(F; L)$, μ_Y captures the degree to which the information learned by F *about* Y is explained by L – whereas $I(F; L)$ only captures the correlation of F and L, regardless of whether this correlation is useful for predicting Y or not. For example, consider if $F(x) = L(x) \cdot Bernoulli(p)$. That is, F is a linear classifier L with noisy outputs. Here, $I(F; L) \ll 1$, due to the noise in F. However, $\mu_Y(F; L) = I(F; Y)$, and thus our metric recovers the fact that all the performance of F in predicting Y is explained by the linear L. Second, Reviewer 3 describes a scenario where F first learns to classify the examples that are incorrectly classified by a linear model L, and notes that μ treats this the same as learning the correct portion of L. This isn't a problem, though, because the incorrect portion of L is exactly the correct portion of the classifier $1 - L$, which is also linear. Contrast this with the following scenario: the samples come from a *mixture* of L and an uncorrelated nonlinear model N, and F learns N first. This is a true example of F not learning the linear part of the distribution, and accordingly $\mu_Y(F, L)$ will equal zero. We will include formal examples to build intuition for our metric in the final version.

[1] On the Spectral Bias of Neural Networks (2018). By Rahaman, Baratin, Arpit, Draxler, Lin, Bengio, Courville.
[2] Understanding training and generalization in deep learning by Fourier analysis (2018). By Zhi-Qin John Xu.
[3] Understanding intermediate layers using linear classifier probes (2016). By Guillaume Alain and Yoshua Bengio.

[1] For Equation (5) in `https://arxiv.org/pdf/1808.04295v4.pdf`, applying Parseval's Theorem.