R1: **The experimental setup** of our paper is designed to both corroborate our theoretical results which are non-trivial, while also demonstrating the possible application of the hypernetwork induced prior in some practical use cases. We concede that our experiments are by no means thorough enough to consider the hypernetwork kernels as a go-to algorithm for image completion tasks, however we feel they do serve their purpose demonstrating the usefulness of hypernetworks induced priors, explaining the validity and inductive bias of the architecture. **Size of f:** While in some of the hypernetworks in the literature f is not very large, this is the case in many of the recent networks, e.g., [B,C,D,E]. **R2:** **We accept the suggested terminology** and would use "hypernet" to refer to network f only. **There are two types of hypernetwork** architectures (including the hypernetwork f and the primary network g). Type A has a much larger network f than g, and the input to f is larger than the input to g. Type B has a smaller f and larger g. In the references that the reviewer mentioned, where one optimizes the input of f, it is more natural to use type B (smaller inputs). However, type A is at least as prevalent in the literature as type B. Examples of type A include cases where g is a single convolutional layer in a deeper network [17,15,5,A]. This is also very prominent in recent work in which the perceptual task is done by a resnet f and the solution is parameterized by a small network g [B,C,D,E]. In such cases, f observes the entire context, while g is a local model, see also [F]. Our analysis holds also for type B networks, given that the network f is wide enough to be approximated by a GP. However, since type B networks are used to find optimal hyperparameters, they require training by definition, and NTKs, which are studied at initialization, are less relevant. Indeed, there is currently no theoretical machinery for understanding the dynamics of optimizing the input with a fixed, trained network in the NTK regime. Our work here presents a first step at a new understanding of hypernetworks through the new machinery of NTK, which covers the type A scenario. We hope to extend this analysis to type B scenarios in the future as well by using the recent Tensor Programs framework. However, this is out of the scope of this contribution. **The MNIST experiment** is a typical type A hypenet setting. The perceptual input is processed by f, and g is a model of the "scene". It directly follows [B,E] (E is a paper R4 pointed to as an example for realistic settings). The reviewer's suggestion to condition f on the digit label is equivalent to learning 10 different denoising networks, which is not utilizing the full power of hypernets. **The computational advantage (L 208)** is meaningful when compared to other kernel methods. From the composition of the hyperkernel (Eq. 12), $\Theta^f(x, x')$ can be evaluated separately for all pairs x, x', instead of evaluating kernel values for all pair of tuples $(x, z), (x', z')$ when considering other kernel methods. When f is a convnet, this can represent a significant reduction in computational cost. **R3:** **Our theoretical results are non-trivial.** In particular, Thm. 1 provides the asymptotic behavior of high order NTK terms which hold for ReLU hypernetworks, as well as regular ReLU MLPs. Our contribution here is both a technical novelty (in the proof) and the significance of the final result. On the technical level, as noted in remark 1 and in L 280-283, we have proven (and improved upon) a conjecture on the asymptotic rates of various correlation functions arising in neural network dynamics (see [5]). As for the result itself, we are the first to arrive at these tight bounds which relate to both hypernetworks and MLPs. Thms. 2 and 3 describe the conditions in which GP behavior emerges in hypernetworks (again nontrivial), and describe the composition of the GP and NTK kernels. We feel these theoretical results are of interest to the community. **The case of a finite f and an infinite g** is left for future work. Note that an infinite g would require f to output an infinite number of parameters. We do discuss the case of both f and g being infinite in Sec. 4. **Reporting variance** we regret not reporting error bars, which will be added. The results were averaged over 10 different training and test splits. **HN outperforms HK in Tab. 2** For small data regimes, the HK outperforms, while for larger datasets, the HN is better. This is consistent with prior observations that kernel methods with NTK tend to outperform in low data regimes. In **Fig. 2**, the input of f is depicted in Row 2, the input of g is a pixel coordinate. The output of g is the pixel intensity in the corresponding coordinate. **Typos** We apologize for the typos and would provide more background on NTK and GP. **R4:** The paper is theoretically driven. **The experiment setup** is very similar to that of the mentioned paper (arXiv 1902.10404) only done on MNIST. To demonstrate this, Fig. I has interpolation results similar to that paper by interpolating between $[\Theta^h(u, u_1), .., \Theta^h(u, u_N)]$ and $[\Theta^h(v, u_1), .., \Theta^h(v, u_N)]$ for two images u and v in Eq. 16. **We regret the lack of details** in the experimental section. The hypernet f in our setup is a convolutional neural network operating on sparse images containing the context points, similarly to [G].

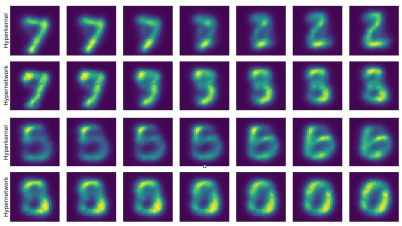

Figure I: Interpolation between 7 ->2 and 5 ->6 using hyperkernel (rows 1 ,3) and hypernetwork (rows 2,4). Both methods used merely 200 samples for training. The hypernetwork trained with sgd clealy underperforms in this low data regime

REFERENCES: **[A]** Wu et al. Pay Less Attention with lightweight and Dynamic Convolutions. ICLR 2019. **[B]** Littwin et al. Deep Meta Functionals for Shape Representation. ICCV, 2019. **[C]** Rotman et al. Electric Analog Circuit Design with Hypernetworks and a Differential Simulator. ICASSP, 2020. **[D]** Bergman et al. Implicit neural representations with periodic activation functions. arXiv:2006.09661, 2020. **[E]** Klocek et al. Hypernetwork functional image representation. ICANN, 2019. **[F]** Nachmani et al. Hyper-Graph-Network Decoders for Block Codes. NeurIPS, 2019. **[G]** Sitzmann et al. Implicit Neural Representations with Periodic Activation Functions. arXiv:2006.09661, 2020.