Investigating Neural Architectures by Synthetic Dataset Design

Adrien Courtois*, Jean-Michel Morel, Pablo Arias
Centre Borelli, ENS Paris-Saclay
4 Av. des Sciences, 91190 Gif-sur-Yvette, France
adrien.courtois@ens-paris-saclay.fr

Abstract

Recent years have seen the emergence of many new neural network structures (architectures and layers). To solve a given task, a network requires a certain set of abilities reflected in its structure. The required abilities depend on each task. There is so far no systematic study of the real capacities of the proposed neural structures. The question of what each structure can and cannot achieve is only partially answered by its performance on common benchmarks. Indeed, natural data contain complex unknown statistical cues. It is therefore impossible to know what cues a given neural structure is taking advantage of in such data. In this work, we sketch a methodology to measure the effect of each structure on a network's ability, by designing ad hoc synthetic datasets. Each dataset is tailored to assess a given ability and is reduced to its simplest form: each input contains exactly the amount of information needed to solve the task. We illustrate our methodology by building three datasets to evaluate each of the three following network properties: a) the ability to link local cues to distant inferences, b) the translation covariance and c) the ability to group pixels with the same characteristics and share information among them. Using a first simplified depth estimation dataset, we pinpoint a serious nonlocal deficit of the U-Net. We then evaluate how to resolve this limitation by embedding its structure with nonlocal layers, which allow computing complex features with long-range dependencies. Using a second dataset, we compare different positional encoding methods and use the results to further improve the U-Net on the depth estimation task. The third introduced dataset serves to demonstrate the need for self-attention-like mechanisms for resolving more realistic depth estimation tasks.

*This work was supported by grants from Région Ile-de-France.

1. Introduction

Deep learning has been characterized by significant advances in fields ranging from computer vision [35] to protein structure prediction [23]. However, neural networks lack interpretability, and it is nearly impossible to predict the performance of a given structure on a task. While most of the effort is directed towards the explainability of the models themselves, the possibility that a better understanding of deep learning methods could come from better designed datasets has received little attention. In this work, we investigate this hypothesis by introducing a methodology to enhance the impact of architectural choices and to identify their flaws.

Datasets of natural images need to be huge in order to capture the semantic complexity of the real world. While such datasets are necessary to ensure generalization to real world applications, their structure and information content is fully out of control. The information given to the network can be ambiguous, sometimes contradictory, and the spatial interaction of features can be guided by hidden statistical dependences. It is therefore hard or impossible to anticipate or explain the success or failure of a given network structure. A second ambiguity resides in the fact that each datum might not contain enough information to solve the prescribed task. A third ambiguity resides in the input itself: a plethora of semantic local and nonlocal cues coexist within the same image, which makes it difficult for an external observer to pinpoint the cause of success or failure of a given network structure.

A better understanding of neural networks requires characterizing their capabilities and linking them to their structure. To this end, we propose to train neural networks on datasets where those ambiguities have been lifted. That way, the success of a structure on a given task can only be attributed to the structure having a certain property, and not to some other uncontrolled statistics. Alleviating the three sorts of ambiguities requires resorting to synthetic datasets. In this work, we introduce a methodology to design such unambiguous synthetic datasets to explore the properties of neural networks.

We illustrate this methodology on three datasets. First, we design a depth estimation task - the Rectangle Depth Estimation (RDE) dataset - to assess the non-local properties of the U-Net which, according to several authors [37, 55], seems to be unable to exploit its large receptive field. In particular, we find that endowing the U-Net with nonlocal layers helps improve its nonlocal capability, especially when a variant of the Lambda layer [4] is used. Then, observations of the failure cases of the resulting structure raise the question of the positional encoding used within the Lambda layer. This leads us to design a second dataset, which aims at assessing the properties of the positional encoding. This second task allows us to design a better positional encoding method, which we successfully transfer to the first task. Finally, we design a dataset to evaluate the ability to group pixels with the same characteristics and share information among them. In particular, we find that self-attention [46] excels at this task.

The contributions of this paper are as follows:

1) We introduce a methodology to design synthetic datasets to be used to evaluate networks' properties. This allows to investigate neural architectures to better understand their capabilities. This methodology can be applied on any structure and for any data modality.

2) We apply this methodology to evaluate three different network properties, namely: the ability to link local cues to distant inferences, the translation covariance and the ability to group pixels with the same characteristics and share information among them. For each property, a dataset is designed. These datasets can be used to evaluate any structure.

3) The datasets are used to compare and discuss multiple structures. The first dataset allows us to find a nonlocal deficit in the U-Net and to partially fix it by adding nonlocal layers in its structure. Then, the second dataset helps us find a way to incorporate positional encoding in the Lambda layer while ensuring translation equivariance. Finally, experiments on the third datasets point out that self-attention and variants excel at grouping pixels with the same characteristics and share information among them. The conclusions we draw on structures might lead to some improvement when handling real datasets, but this is not the goal of this paper. Rather, the proposed methodology may be used to verify unambiguously the effect of each proposed neural structure on *ad hoc* synthetic data.

All of our results can be reproduced in less than one day on a single GPU. Both the code and dataset are available on GitHub.

2. Related work

Multiple depth estimation datasets [8, 28, 50, 53] aim at training networks for real applications. The RDE dataset we introduce is a depth estimation task reduced to its simplest form, where only the strictly necessary cues are left for the network to understand the depth ordering of the scene. Other synthetic datasets [36, 44] have been proposed to analyze and quantify the effect of certain layers or training methods, allowing one to discover effects that would otherwise be impossible to unveil. Notably, synthetic datasets are commonly used for image quality evaluation [26]. The Long-Range Arena [44] was introduced to evaluate the long-range capabilities of Transformers [46]. While we share a similar objective, failure on such complex classification tasks cannot be easily linked to structural deficiencies. We aim at designing synthetic datasets for better understanding structures and not only assessing them. The Color Code dataset is used to assess variants of Transformers, but both the RDE and the Centered Square dataset consist of images of small size but too large for the quadratic cost of Transformers. The RDE dataset shares similarities with the dead leaves model [16] and builds images composed of rectangles to create occlusion.

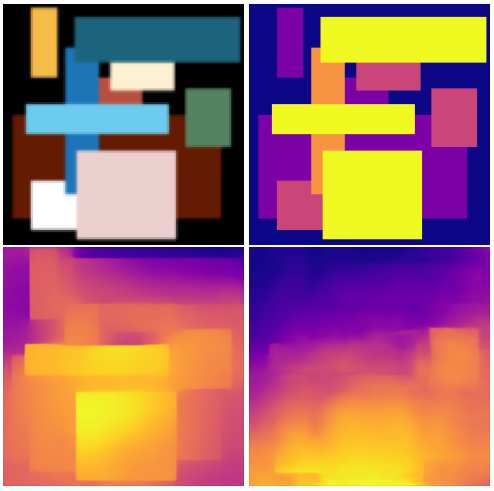

Figure 1: Results of state-of-the-art networks on our depth estimation problem, without retraining. First line: input and ground truth. Second line: result of MiDaS [38], result of MergeNet [32]. The disappointing results of SOTA networks on a visually unambiguous image show that these networks are guided by hidden natural statistics, much more than by nonlocal geometric reasoning.

In particular, the approach described in [29] is close to ours. The authors exhibit a property they want their network to have and design simple synthetic datasets to evaluate it. They find that their network does not have the property and propose a change in structure to solve the issue. In this work, we propose a generalization of this approach by providing a methodology to reproduce those steps for other properties. In [25], the authors also introduce a dataset to exhibit a property their layer has, but competing approaches do not. It can be argued however that the usage of noise

introduces unneeded information and does not follow the Occam razor criterion.

Different ways to improve the U-Net [39] have been proposed in the literature. For instance, U^2-Net [37] provides a global receptive at each scale by including a U-Net at each scale of the U-Net. This method is state-of-the-art for figure-background segmentation. In [2] the receptive field and the amount of processing are increased by a recurrent network used at each scale of the encoder. We shall actually propose here a faster and lighter-weight approach by leveraging non-local layers to attain global receptive field at each scale. In [55] the authors identify a receptive field issue with the U-Net. They propose to solve it by a novel structure processing all scales in parallel. The LambdaUNet [34] uses the Lambda layer [4] in conjunction with the U-Net. It keeps the Lambda layer in its local formulation, while we shall change its receptive field to cover the entire image at once. Notably, the authors of [49] introduced a network called "Non-Local U-Net". They use a so-called "non-local layer" [48] similar to self-attention [46] to increase the receptive field of the U-Net. The resulting network is slow, as it is based on an operation with quadratic time and space complexity. In comparison, we shall explore a U-Net architecture that can be combined with a variety of non-local layers. The layers we choose to assess have a linear time and space complexity and can be trained and evaluated on a single GPU.

A wide variety of non-local layers have been proposed in the literature. Many of them are based on self-attention or "non-local networks" [48]. Some layers aim at mimicking self-attention with a linear complexity [47, 41, 11, 52, 24, 22, 20, 54, 51]. We evaluated some of those layers but were not able to make them converge on our task, or they were exceedingly long to train. This suggests that they require heavy hyper-parameter tuning or the usage of multiple convergence tricks. In this work, we explicitly chose to assess easy-to-use and easy-to-train layers. Other non-local layers [4, 5, 10, 12, 18, 19, 42, 56] do not try to mimic self-attention. Any of them could be incorporated in our architecture. We shall evaluate several of them.

Since its first introduction in [46], different approaches have proposed different positional encoding methods. In [6], it is pointed out that the positional encoding in its original formulation is not translation covariant. The authors propose to decorrelate the encoding of the absolute position with the encoding of the relative position. Their findings suggest that relative position alone is enough for some tasks in NLP. The original position encoding is a predefined sinusoidal function, and some works have focused on improving these functions [30, 14, 27]. Other approaches have been developed, see [15] for an overview. In this work, we use the Centered Square dataset to evaluate different positional encoding methods to be used inside the Lambda layer to ensure translation covariance.

3. The methodology

In this section, we describe the design requirements an unambiguous dataset must fulfill to assess whether or not a structure has a given property such as nonlocality, translation covariance, *etc*. Such requirements can only be fully enforced in a synthetic dataset, namely:

Unambiguous ground-truth: There must be no contradictory labels, no annotation problems nor cases where multiple labels are valid for the same input.

Well-posedness: The input contains enough information to solve the task. There must exist a reconstruction algorithm able to deduce the exact ground truth from the input image. In other terms, reaching 100% accuracy is theoretically possible. Note that, because of the inherent ambiguity of natural scenes, this property is not attainable with natural datasets.

Focus on a specific network's property: The network must be able to deduce the exact ground truth from the input image only if it has the assessed property (such as nonlocality, permutation invariance, etc.). So the cues given to the network must be under full control, so that we know exactly which cues the network can use. Note again that this property is not attainable with natural datasets, as they contain many statistical cues that help compensate for a structural deficiency of the network.

In particular, we would like to stress out that an important requirement is *simplicity*. The third property can only be enforced if the dataset is as simple as possible.

In the following, we describe the three datasets we used as an illustration of our methodology for, respectively, *nonlocality*, *translation covariance* and *the ability to pass on information to every pixel of the same color*.

3.1. The Rectangle Depth Estimation dataset

The dataset consists in a depth estimation task where objects are replaced by simple rectangles. The rectangles can overlap and occlude one another, creating a spatial organization that naturally puts objects of top of others. To compute an unambiguous ground truth, our reconstruction algorithm is based on three nonlocal cues: a) color similarity (all rectangles are monochromatic, thus can be recovered nonlocally); b) T-junctions, a local cue that propagates nonlocally; c) convexity, that leads to decide that a region occluded by another shape is to be inpainted as a convex shape and is therefore underneath the occluding shape. A full description of the algorithm is available in the supplementary materials. An example can be found in Figure 2.

The task that needs to be solved is closely related to real-world depth estimation as it accurately reflects its main difficulties. When for example an object is partially occluded

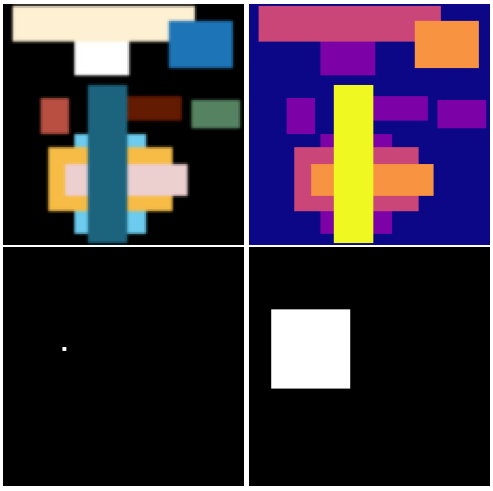

Figure 2: Top row: An example of image of the RDE dataset and the associated ground truth. The brighter the color, the higher the number of rectangles that are beneath it. Bottom row: An example of image of the Centered Square dataset and the associated ground truth with $H = W = 64$ and $w = 21$.

by others, it is divided into several components and the network must regroup the separated parts. This can only be done by recognizing the same color and/or detecting edge alignment.

To decide about the depth ordering, the network can only rely on T-junctions and convexity. These local cues need to be successfully detected and propagated at an arbitrary distance for understanding the geometrical organization of the scene. Indeed, the distance can be arbitrarily large as there is no upper limit on the size of the rectangles. Therefore, the network cannot overfit on local information, whereas in natural scenes it is easy for a network to differentiate, say, a tree from the background sky. Whilst these could be interesting priors, there is no guarantee that the network will not associate a depth value to each position or texture. Our dataset is designed so that it is not possible to associate a local patch to an absolute depth. All cues provide information about relative ordering between objects. The global depth can only emerge via a coherent global integration of these relative cues. In Figure 1 we show an example where state-of-the-art networks trained on natural images seem to heavily rely on local cues and natural statistics. We can for instance see that the bottom of the picture always seems to be brighter than the top, even though it makes no sense in this case. Of course, these networks being trained on natural data, it was to be expected that they would perform poorly on work on our out of domain dataset. Nevertheless, this experiment is interesting because it shows statistical priors on the depth learnt. In our dataset, a failure cannot be attributed to a misunderstanding of the objects caused for instance by poor lightning conditions or noise. The dataset being fully unambiguous and its ground truth recoverable from geometric features in sight, failure can only be attributed to a poor geometrical understanding. This allows one to assess the ability for a network to compute non-local features (or, for the case of the U-Net, to efficiently use the multi-scale structure). This also suggests that any improvement on this dataset should be reflected on other depth estimation datasets.

3.2. The Centered Square dataset

This dataset is designed to assess the translation covariance of a given positional encoding method. We use it to find the best method to use inside our Lambda layer. The input consists of an all-black $H \times W$ image where a single pixel is white. The associated ground truth is an all-black image with a white square of width w centered around the white pixel. The training set consists of all the positions for the white pixel contained in the square of dimension $\frac{H}{2} \times \frac{W}{2}$ located in the center of the image. The test set is composed of all the other positions, except for the ones where the image's boundaries crop the ground truth square. This way, the network only learns the reconstruction property in the middle of the image and is evaluated on its ability to apply this property everywhere in the image. A network can only do it perfectly if it is translation equivariant. An example of input and label is shown in Figure 2.

3.3. The Color Code dataset

One of the limitations of the RDE dataset is that it uses 10 fixed colors for the entire dataset *i.e.* for every image, the 10 same colors are used to color the rectangles. We made this choice so the network could focus on the non-local reasoning, even if it implies overfitting on the fixed colors to overcome occlusion. In particular, we found that the baseline U-Net is not able to overcome occlusion even in this simplistic scenario. In our attempt to progressively bridge the gap between synthetic and real depth estimation, the next natural step is to change the colors for each image. In this scenario, a network must solve two tasks: first, it must use local and nonlocal cues to find a mapping between color and depth and secondly, it must pass on this depth to every pixel of this color.

As the second task is difficult in itself, we decided to study the performance of different layers on this task alone. This leads us to the introduction of the Color Code dataset. This third dataset aims to study the performance of a network which, given a mapping between colors and codes, must pass to every pixel the code corresponding to its color. More formally, for each input k colors $c_1, \dots, c_k$ are randomly sampled. For each color c_i, a code z_i is randomly picked. Then, a mapping $\sigma : [\![1, k]\!] \to [\![1, N]\!]$ is sampled as

well as a mask $m \in \{0,1\}^N$ such that the input is given by

$$x = \begin{pmatrix} c_{\sigma(1)} & \dots & c_{\sigma(N)} \\ m_1 \cdot z_{\sigma(1)} & \dots & m_N \cdot z_{\sigma(N)} \end{pmatrix},$$

and the associated ground truth is

$$y = \begin{pmatrix} z_{\sigma(1)} & \dots & z_{\sigma(N)} \end{pmatrix}.$$

In other terms, for some positions the code is given and for others it is not. The goal of the network is to find where the code associated with a color has been given, retrieve it and propagate it to the right positions.

4. Non-Local U-Net

Our baseline for the depth estimation problem is a traditional U-Net [39] with concatenated skip connections. To limit the number of parameters, we kept the width at each scale constant and equal to 48 instead of doubling it after each down-sampling. In accordance with [43], we observed that this alleviated overfitting in multiple scenarios. The resulting network had 871 729 parameters. With the multiple skip connections and the hourglass structure, the U-Net is known to be stable to a variety of learning rates and training schedules. We modified this U-Net following the recipe of [31]. This includes using GELU [17], LayerNorm [3] instead of BatchNorm [21], 7×7 grouped convolutions, the inverted bottleneck structure after each block [40] and LayerScale [45].

To host nonlocal layers, we passed the input feature map at each scale into a local module and a nonlocal module. The two outputs were then concatenated, upsampled/downsampled and passed on to the next scale. The local module corresponds to the module of the original U-Net and the nonlocal module is the nonlocal layer. We refer the reader to Appendix A for further details.

All the networks we considered have five down-sampling operations. The smallest feature map has a 4×4 spatial extent. Therefore, the receptive field of the baseline U-Net and all of its considered variants cover the entire image.

We tried four different non-local layers: the Lambda Layer [4] (which is itself a variant of [9]), the Global Context Layer [5], Global Average Pooling, Deformable Convolutions [56]. We chose these over others because they could be applied at each scale and fit on a single GPU.

When it was not already the case, we embedded the nonlocal layer with a PreNorm [33], skip connections and an inverted bottleneck structure to process its output. We also fixed stability issues when discovered. We found that these simple tweaks led to stabler convergence and better results. We modified the original Lambda layer to avoid using its positional encoding, which has a quadratic time/space complexity. More details can be found in Section and the supplementary.

5. Experiments

5.1. Experiments on the RDE dataset

5.1.1 Metrics

We used three of the most commonly employed metrics for Monocular Depth Estimation tasks [1, 7, 32, 38]: the Root Mean Square Error (RMSE), the $\delta_{1.25}$ and the Ord metric. We also used the generalization gap as an indicator of how well the assessed networks generalize [13]. The RMSE was defined by

$$\text{RMSE}(\hat{y}, y) := \sqrt{\frac{1}{HW} \sum_{i,j} (\hat{y}_{i,j} - y_{i,j})^2},$$

where $\hat{y}$ is the prediction and y the ground-truth. The percentage of pixels with $\delta_{1.25}$ is given by

$$\delta_{1.25} := \frac{1}{HW} \sum_{i,j} \mathbf{1}_{\{\max\left(\frac{\hat{y}_{i,j}}{y_{i,j}}, \frac{y_{i,j}}{\hat{y}_{i,j}}\right) > 1.25\}}.$$

The ordinal loss consists in sampling 50,000 pairs of pixels $((i_1, j_1), (i_2, j_2))$ and for each of those pairs, compute:

$$l_i = \begin{cases} +1, & \text{if } y_{i_1,j_1} / y_{i_2,j_2} \geq 1 + \tau \\ -1, & \text{if } y_{i_1,j_1} / y_{i_2,j_2} \leq \frac{1}{1+\tau} \\ 0, & \text{otherwise.} \end{cases}$$

Using the same pairs, the equivalent quantity $\hat{l}$ is computed for the prediction. The ordinal loss is given by:

$$\text{Ord} := \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \mathbf{1}_{\{l_i \neq \hat{l}_i\}}.$$

Finally, we define the generalization gap as the difference between the value of the loss on the test set and on the train set at the end of the training.

In practice, we used $\tau = 0.03$ and all the networks were evaluated using the same set of pairs of pixels when computing the ordinal loss.

5.1.2 Results

Effect of ambiguity removal We trained the Non-Local U-Net with different nonlocal layers on the RDE dataset. This dataset was comprised of images of dimension 128×128. Most images featured 10 rectangles. The dataset was filtered so as to remove most ambiguous cases, *e.g.* when T-junctions are hidden by another square or when rectangle sides are aligned. As an illustration of the need for an unambiguous dataset, we compare in Table 1 the performance of the baseline network when trained on the unambiguous dataset and when trained on the same dataset but where we did not remove the ambiguous cases. The network trained on the unambiguous dataset is four times better than its counterpart.

Comparison We report in the upper part of Table 1 the results of the different assessed nonlocal layers on the RDE dataset. They show the Lambda layer yielding the best performance for most metrics.

The deformable convolutions yielded the lowest performance. This is most likely due to the fact that it has the smallest width. Since this layer introduced a large number of parameters, we had to reduce the width so it had a number of parameters close to the baseline. Its width was 21 when most layers had around 40 channels per feature map.

Overall, even the simplest non-local layer yielded a noticeable improvement over the baseline U-Net. This supports the claim of [37] and [55] that the U-Net might be more local than expected. It seems that the more sophisticated the non-local layer, the better the results, which suggests that further improvement could come from still better nonlocal layers.

Although the U-Net has a global receptive field, the way the information propagates inside it might be to blame. This information is fused locally, step by step, in the way of a diffusion process. This might explain why occlusions stop the propagation of information from a piece of an occluded object to another, as can be observed in Figure 3. See Section 6 for more details along with an illustration.

When observing the cases where our best network failed, we observed that the network struggled in the case the T-junctions between two rectangles are occluded. In this case, the network needs to compute the spatial extent of each rectangle from the visible parts and understand that the extensions overlap. See Figure 4 for an illustration. Failure to handle such case suggests a problem with the positional encoding used within the Lambda layer, which leads us to the Centered Square dataset.

5.2. Results on the Centered Square dataset

For this set of experiments, we used the Centered Square dataset presented in Section 3.2 with $H = W = 64$ and a square width of $w = 21$. The dataset was composed of 484 training images and 1,452 test images. We evaluated our network by computing the IOU over the test set. Further training details are given in the supplementary.

As the goal of this dataset was to evaluate the positional encoding method, the network to be trained was reduced to its simplest form. Indeed, using a multiscale structure could bias the interpretation of the results. On this task, we trained a network made of one 1×1 convolution, followed by the Lambda layer using the positional encoding method being investigated and by another 1×1 convolution.

In NLP, the Transformer-based architecture almost exclusively relies on positional encoding strategies to encode the relative and absolute positions of words in a sentence. The original Lambda layer is inspired by the Transformer architecture but the original positional encoding of the lambda layer can be very costly in terms of both parameters and computations. Therefore, we decided to replace it with the cosine positional embedding presented in [46]. Our first approach was to simply add the positional encod-

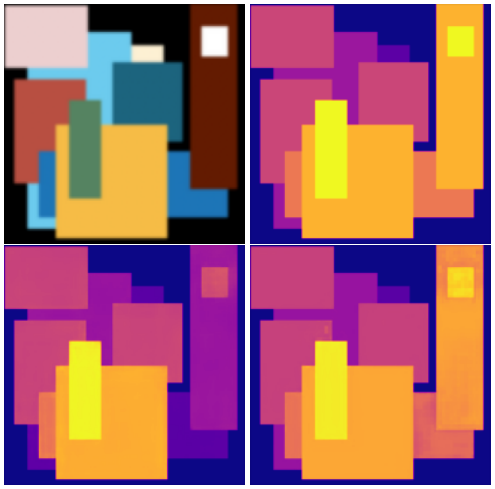

Figure 3: An example of case where the U-Net without nonlocal layers is not able to overcome occlusion. First line: input, ground truth; second line: output of the baseline U-Net, output of the Non-Local U-Net + Lambda + PE. To solve this case, the network must propagate the depth information it found on the left of the shape to the rest of the shape, with the help of the information of color.

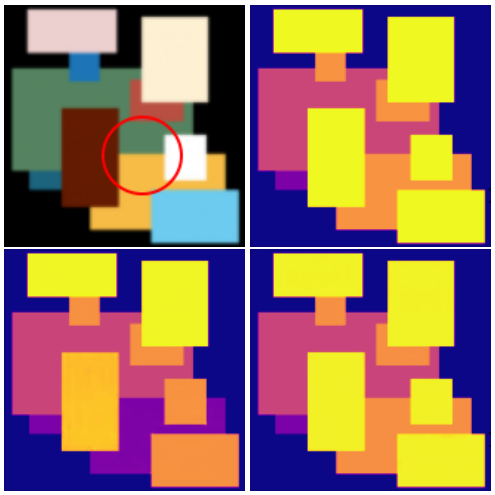

Figure 4: An example of case where the T-junctions between two rectangles are occluded. First line: input, ground truth; second line: output of the Non-Local U-Net + Lambda, output of the Non-Local U-Net + Lambda + PE. To solve this case, the network must compute the spatial extent of the occluded rectangles and determine which is on top. Incorporating a translation covariant positional encoding in the Lambda layer partially solved these problems.

Network	# parameters	Test loss ↓	Gen. gap ↓	Ord ↓	$\delta_{1.25}$ ↓	RMSE ↓
Baseline*	871 729	1.78	-	6.03	7.36	4.91
Baseline	871 729	0.72	0.36	1.79	1.98	1.95
Global Context	864 477	0.54	0.32	1.09	1.39	1.49
Global Average Pooling	883 049	0.60	0.31	1.47	1.51	1.71
Deformable	864 844	0.94	0.54	2.43	2.83	2.44
Lambda	871 945	<u>0.28</u>	**0.14**	**0.47**	<u>0.64</u>	<u>0.82</u>
Lambda + PE	871 945	**0.25**	<u>0.15</u>	<u>0.50</u>	**0.58**	**0.77**
Lambda + PE + TT	928 969	0.28	0.18	0.59	0.65	0.85

Table 1: Results on the RDE dataset. All metrics are multiplied by 100 for readability. The best reported results are in bold and the second best is underlined. Note that the more sophisticated the non-local layer, the better the results. The model marked with an asterisk (*) was trained on the ambiguous version of the training dataset and evaluated on the unambiguous one. In particular, the reported test loss is the one computed on the unambiguous dataset.

ing to the input feature map and pass the result through the Lambda layer. As pointed out in [6], this leaks the absolute position of the pixel, which could be detrimental for our task. We therefore decided to decorrelate the positional encoding from the rest of the layer by adapting the method of [6] to the lambda layer. Formally, given a predefined positional encoding $P \in \mathbb{R}^{C\times N}$, an input feature map $x \in \mathbb{R}^{C_{\text{in}}\times N}$ and learnable matrices $K \in \mathbb{R}^{M\times C_{\text{in}}}$, $V \in \mathbb{R}^{C_{\text{out}}\times C_{\text{in}}}$, $Q \in \mathbb{R}^{M\times C_{\text{in}}}$, $A \in \mathbb{R}^{C_{\text{out}}\times M}$ we compute

$$\bar{K} = \text{SOFTMAX}_N(Kx) \in \mathbb{R}^{M\times N},$$

$$\lambda_{\text{content}} = \bar{K}(Vx)^T \in \mathbb{R}^{M\times C_{\text{out}}},$$

$$\lambda_{\text{pos}} = A\bar{K}P^T \in \mathbb{R}^{C_{\text{out}}\times C},$$

$$y_{\text{pos}} = \lambda_{\text{pos}} P \in \mathbb{R}^{C_{\text{out}}\times N},$$

$$y_{\text{content}} = \lambda_{\text{content}}^T Qx \in \mathbb{R}^{C_{\text{out}}\times N},$$

and the output of the Lambda layer is given by $y = y_{\text{content}} + y_{\text{pos}}$. We refer to this method as "Decor.". Finally, we noted that the cosine positional encoding of [46] is of the form

$$P_{c,n} = \begin{cases} \cos(w_k n) & \text{if } c = 2k \\ \sin(w_k n) & \text{if } c = 2k+1 \end{cases},$$

and we investigated if another choice of the sequence $(w_c)_{c\in[\![1,C/2]\!]}$ could yield better results. This led us to introduce Fourier coefficients $w_c = 2\pi c/(2C)$, $c = 1, \ldots, C/2$. We refer to this method as "Fourier".

In Table 2, we compare these methods with the use of CoordConv [29] instead of regular convolutions within the Lambda layer, and with the translation covariant version of the positional encoding proposed in the original Lambda layer with different widths R. Notably, we found that the only truly translation covariant approaches were the one that used the "Decor." mechanism. In particular, using the Fourier positional encoding alongside the "Decor." mechanism yields a perfect score on the dataset. We tested our variant of the Lambda layer with this new positional encoding method (Lambda + PE) on the RDE dataset. This modification moderately improved on the final performance as reported in Table 1.

5.3. Results on the ColorCode dataset

For this set of experiments, we used the Color Code dataset presented in Section 3.3. The input dimension was $N = 128$, the number of different colors per input $k = 10$ and the proportion of masked inputs 50%. We trained our networks with 20,000 training images and evaluated them on a separate test set comprised of 10,000 testing images. The results are presented in Table 3.

The spatial dimension of the input being low, we chose to evaluate the smallest versions of the linear-cost approximation of the self-attention. As we only needed to investigate layer capabilities, we reduced the trained networks to their simplest form: a 1×1 convolution, followed by three instances of the assessed layer, followed by another 1×1 convolution. See the supplementary for more details.

The used metric was the mean accuracy of the masked codes across the training dataset *i.e.*

$$\frac{\sum_{i=1}^{N}(1-m_i)\cdot \mathbf{1}_{\{z_{\sigma(i)}=\hat{z}_i\}}}{\sum_{i=1}^{N}(1-m_i)},$$

where $\hat{z}$ is the network's prediction.

Notably, all self-attention variants perform on par. The overall performance of the Transformer suggests that some ambiguity was left in the dataset. This ambiguity was probably due to some colors not being easily distinguishable. Humans would also fail in some cases as it is hard to tell apart the 256^3 different colors. The failure of the MLP-Mixer on this task seems to indicate that multilayer perceptrons are not always a good replacement for self-attention, even if they are on image classification.

Method	# parameters	Train IOU	Test IOU
Cosine + Sum + QKV	182 018	<u>98.70%</u>	24.71%
Cosine + Sum + QV	182 018	98.67%	35.66%
Cosine + Decor.	182 018	84.78%	<u>84.79%</u>
Fourier + Decor.	182 018	**100.0%**	**100.0%**
CoordConv	184 066	80.71%	56.16%
Lambda ($R = 7$)	182 818	11.11%	11.11%
Lambda ($R = 19$)	187 810	19.50%	14.81%

Table 2: Results on the Centered Square dataset with input dimension $H = W = 64$ and square width $w = 21$. The best reported results are in bold and the second best are underlined. The "Decor." method is the most consistent as it performs almost identically on the train and test sets.

Network	# parameters	Test metric
Transformer	2 373 379	**99.87%**
MLP-Mixer	1 979 011	69.77%
Nyströmformer-32	2 176 003	99.66%
Linformer-32	2 208 771	99.15%
Reformer-32	2 176 003	77.59%
Lambda	2 175 235	99.52%
Lambda + TT	2 569 987	<u>99.85%</u>

Table 3: Results on the Color Code dataset with $N = 128$ positions and $k = 10$ different colors for each input. In particular, the Transformer seems to have reached the maximum possible metric on this dataset.

The only attention map computed in the Lambda layer depends on the comparison between the input feature map and a learned matrix. Inspired by the mechanism of self-attention, we improved this attention map by switching the learned matrix for a matrix computed based on the input feature map. This amounts to iterating the Lambda layer **twice**, yielding our variant named *Lambda + TT*. We refer the reader to the supplementary for further details. On the Color Code dataset, this slight modification of the Lambda layer yields the second best performance. We tested this layer onto the RDE dataset and surprisingly, we found a decrease in performance as reported in Table 1. The lack of improvement could be due to this additional mechanism being not needed since there are only 10 colors in the entire dataset. It could also be due to the fact it was trained for 50 epochs while it could have benefited from a longer training.

6. Limitations, conclusion and future works

6.1. Limitations

This work is about better understanding the properties of neural structures. The goal of the methodology is to pinpoint the properties of each given structure. To this end, we remove all the complex unknown statistical cues inherent to natural images. The final goal is that, given a task to solve, a practitioner will first identify the properties needed to solve the task and will choose the components of the network accordingly. This nonetheless raises several issues. First, there is no guarantee that simple properties are easily identifiable for every task. Secondly, there is no guarantee that if we mix multiple structures with different properties, the resulting structure will have the properties of its components. Thirdly, even if we had managed to find a structure with all of the desired properties, it might be that it doesn't transfer to natural images.

Furthermore, all of this work is constrained by the optimization process: it might be that a structure that does not work on a given dataset would yield a very good result if trained using a different training recipe.

6.2. Conclusion and future work

We attempted to design synthetic datasets as tools to compare and improve neural networks. Very controlled datasets like RDE might play the role that was formerly given in signal processing to the impulses that were fed to a black box to obtain an impulse response. Here, the goal is to keep interpretable results that link network structure changes to performance gains. We claim that such interpretations can hardly be obtained with natural annotated datasets.

We plan to expend RDE to more general scenes while keeping its statistical neutrality. The current dataset does not address the non-local problem of detecting the main colors (the ten colors were fixed once and for all in the dataset). We plan to vary the number of rectangles, then to authorize more varied shapes, finally to endow them with textures, so as to keep the synthetic dataset visually interpretable, statistically neutral, but ever closer in complexity to a natural scene.

References

[1] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941*, 2018. 5

[2] Md Zahangir Alom, Chris Yakopcic, Mahmudul Hasan, Tarek M Taha, and Vijayan K Asari. Recurrent residual u-net for medical image segmentation. *Journal of Medical Imaging*, 6(1):014006, 2019. 3

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Hinton Geoffrey E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5

[4] Irwan Bello. Lambdanetworks: Modeling long-range interactions without attention. *arXiv preprint arXiv:2102.08602*, 2021. 2, 3, 5

[5] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Global context networks. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 3, 5

[6] Pu-Chin Chen, Henry Tsai, Srinadh Bhojanapalli, Hyung Won Chung, Yin-Wen Chang, and Chun-Sung Ferng. Demystifying the better performance of position encoding variants for transformer. *arXiv preprint arXiv:2104.08698*, 2021. 3, 7

[7] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. *Advances in neural information processing systems*, 29:730–738, 2016. 5

[8] Weifeng Chen, Shengyi Qian, David Fan, Noriyuki Kojima, Max Hamilton, and Jia Deng. Oasis: A large-scale dataset for single image 3d in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 679–688, 2020. 2

[9] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. A2-nets: Double attention networks. *arXiv preprint arXiv:1810.11579*, 2018. 5

[10] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 433–442, 2019. 3

[11] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020. 3

[12] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 3

[13] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *arXiv preprint arXiv:2106.04803*, 2021. 5

[14] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018. 3

[15] Philipp Dufter, Martin Schmitt, and Hinrich Schütze. Position information in transformers: An overview. *arXiv preprint arXiv:2102.11090*, 2021. 3

[16] Yann Gousseau and François Roueff. The dead leaves model: general results and limits at small scales. *arXiv preprint math/0312035*, 2003. 2

[17] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 5

[18] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. *arXiv preprint arXiv:1810.12348*, 2018. 3

[19] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 3

[20] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019. 3

[21] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning*, pages 448–456, 2015. 5

[22] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021. 3

[23] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021. 1

[24] Young Jin Kim and Hany Hassan Awadalla. Fastformers: Highly efficient transformer models for natural language understanding. *arXiv preprint arXiv:2010.13382*, 2020. 3

[25] Filippos Kokkinos, Ioannis Marras, Matteo Maggioni, Gregory Slabaugh, and Stefanos Zafeiriou. Pixel adaptive filtering units. *arXiv preprint arXiv:1911.10581*, 2019. 2

[26] Debarati Kundu, Lark Kwon Choi, Alan C Bovik, and Brian L Evans. Perceptual quality evaluation of synthetic pictures distorted by compression and transmission. *Signal Processing: Image Communication*, 61:54–72, 2018. 2

[27] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019. 3

[28] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018. 2

[29] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. *Advances in neural information processing systems*, 31, 2018. 2, 7

[30] Xuanqing Liu, Hsiang-Fu Yu, Inderjit Dhillon, and Cho-Jui Hsieh. Learning to encode position for transformer with continuous dynamical model. In *International Conference on Machine Learning*, pages 6327–6335. PMLR, 2020. 3

[31] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022. 5

[32] S Mahdi H Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yagiz Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9685–9694, 2021. 2, 5

[33] Toan Q Nguyen and Julian Salazar. Transformers without tears: Improving the normalization of self-attention. *arXiv preprint arXiv:1910.05895*, 2019. 5

[34] Yanglan Ou, Ye Yuan, Xiaolei Huang, Kelvin Wong, John Volpi, James Z Wang, and Stephen TC Wong. Lambdaunet: 2.5 d stroke lesion segmentation of diffusion-weighted mr images. *arXiv preprint arXiv:2104.13917*, 2021. 3

[35] Sandeep Paul, Lotika Singh, et al. A review on advances in deep learning. In *2015 IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI)*, pages 1–6. IEEE, 2015. 1

[36] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. In *ICLR MATH-AI Workshop*, 2021. 2

[37] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, 106:107404, 2020. 2, 3, 6

[38] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv preprint arXiv:1907.01341*, 2019. 2, 5

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3, 5

[40] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 5

[41] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3531–3539, 2021. 3

[42] Domen Tabernik, Matej Kristan, and Aleš Leonardis. Spatially-adaptive filter units for compact and efficient deep neural networks. *International Journal of Computer Vision*, 128(8):2049–2067, 2020. 3

[43] Mingxing Tan and Quoc V Le. Efficientnetv2: Smaller models and faster training. *arXiv preprint arXiv:2104.00298*, 2021. 5

[44] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006*, 2020. 2

[45] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 32–42, 2021. 5

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2, 3, 6, 7

[47] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 3

[48] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 3

[49] Zhengyang Wang, Na Zou, Dinggang Shen, and Shuiwang Ji. Non-local u-nets for biomedical image segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6315–6322, 2020. 3

[50] Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, Yang Xiao, Ruibo Li, and Zhenbo Luo. Monocular relative depth perception with web stereo data supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 311–320, 2018. 2

[51] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nystöm-based algorithm for approximating self-attention. In *Proceedings of the... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, volume 35, page 14138. NIH Public Access, 2021. 3

[52] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. *arXiv preprint arXiv:2102.03902*, 2021. 3

[53] Haichao Yu, Ning Xu, Zilong Huang, Yuqian Zhou, and Humphrey Shi. High-resolution deep image matting. *arXiv preprint arXiv:2009.06613*, 2020. 2

[54] Chen Zhu, Wei Ping, Chaowei Xiao, Mohammad Shoeybi, Tom Goldstein, Anima Anandkumar, and Bryan Catanzaro. Long-short transformer: Efficient transformers for language and vision. *arXiv preprint arXiv:2107.02192*, 2021. 3

[55] Qing Zhu, Cheng Liao, Han Hu, Xiaoming Mei, and Haifeng Li. Map-net: Multiple attending path neural network for building footprint extraction from remote sensed imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 2020. 2, 3, 6

[56] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019. 3, 5