Efficient learning through compositionality in a CNN-RNN model consisting of a bottom-up and a top-down pathway

Sarah Fabi (sarah.fabi@uni-tuebingen.de)
Neuro-Cognitive Modeling Group, University of Tübingen, Germany

Lena Holzwarth
Neuro-Cognitive Modeling Group, University of Tübingen, Germany

Martin V. Butz
Neuro-Cognitive Modeling Group, University of Tübingen, Germany

Abstract

Learning to write is characterized by bottom-up mimicking of characters and top-down writing from memory. We introduce a CNN-RNN model that implements both pathways: It can (i) directly write a letter by generating a motion trajectory given an image, (ii) first classify the character in the image and then determine its motion trajectory 'from memory', or (iii) use a combination of both pathways. The results show that, in one-shot and few-shot learning, the model profits from different combinations of the pathways: The generation of different character variants works best when the top-down is supported by the bottom-up pathway. Refilling occluded images of efficiently learned characters works best when using the top-down pathway alone. Overall, the architecture implies that a weighted merge of bottom-up and top-down information into a latent, generative code fosters the development of compositional encodings, which can be reused in efficient learning tasks.

Keywords: top-down processing; bottom-up processing; one-shot learning; compositional encodings; efficient learning; RNN

Introduction

Different philosophers and psychologists disagreed for a long time about bottom-up and top-down explanations of human learning (Gopnik, 2019). To name a few, Aristotle, Hume, Mill, Pavlov, and Skinner defended a behavioristic, bottom-up approach that assumes that humans do not need any prior knowledge. Instead, they are supposed to learn simply by extracting associations and patterns from the incoming sensory stream. In contrast, Plato, Descartes, Kant, and Chomsky suggested that humans must have basic knowledge of abstract concepts to be able to form testable predictions and hypotheses in a top-down manner (Gopnik, 2019). Nowadays, most researchers agree that those two strands can be integrated, which is why we expect artificial models of cognition to profit from the inclusion of bottom-up and top-down processes.

When children learn to write, they first try to copy the characters they see. Meanwhile, they learn to classify those characters, such that they can later on write a character from memory. While the first is a bottom-up process, the latter is a top-down process, which generates a trajectory from a compact, internal encoding. Over time, the proportion of both processes may vary from the mere copying to writing from an idea on how a character is usually written, leading to the development of one's own handwriting style. The proportion of bottom-up versus top-down character writing might further vary depending on the task. Even after developing one's own handwriting style, in some occasions, it might be necessary to copy characters, for example, when trying to copy another person's handwriting. In other occasions, such as when recognizing characters that are partially occluded, it is necessary to complete the full character from memory.

This work is based on previous work on the artificial generation of handwritten characters. Particularly the learning of characters in a one-shot manner has been investigated in depth over recent years (Lake, Salakhutdinov, & Tenenbaum, 2019). Fabi, Otte, Wiese, and Butz (2020) have shown how a one-shot inference mechanism can tap into compressed, compositional generative structures in RNN models. The approach was inspired by human cognition, which is able to divide objects into components and to thus rearrange them at will later on when confronted with or when imagining related objects. In particular, Fabi, Otte, and Butz (2021b) investigated the inner workings of generative long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) networks while producing character trajectories, providing evidence that the model was successful in several one-shot tasks because it extracted character components during training that it could reuse when confronted with new characters. In detail, the incorporation of an embedding layer and inverse latent state inference (Otte, Schmitt, Friston, & Butz, 2017; Butz, Bilkey, Humaidan, Knott, & Otte, 2019) enabled the system to flexibly recombine previously learned compositional encodings. As a result, one- and few-shot learning becomes possible, effectively generating handwritten character trajectories out of one-hot encoded inputs.

Because of this earlier success and the demand to include compositional capabilities into machine learning algorithms (Battaglia et al., 2018; Franklin, Norman, Ranganath, Zacks, & Gershman, 2020; Gopnik, 2019; Lake, Ullman, Tenenbaum, & Gershman, 2017), here we investigate whether similar one- and few-shot learning abilities can be elicited in a more complex RNN model that is inspired by human bottom-up and top-down processes. The new model is able to generate handwritten character trajectories out of images. It consists of (i) a bottom-up, direct pathway that mimics the redrawing process in humans and (ii) a top-down pathway, which classifies the image before generating a class-corresponding trajectory. Both routes can be merged at will.

After introducing the full model, we investigate whether the model is able to generate characters in a one-shot manner. With the help of a particular one-shot inference mechanism,

In J. Culbertson, A. Perfors, H. Rabagliati & V. Ramenzoni (Eds.), *Proceedings of the 44th Annual Conference of the Cognitive Science Society.*

we show that it is possible to directly infer a suitable, compact encoding of a novel motion trajectory. We furthermore evaluate the model in the generation of various variants of particular letter concepts or in refilling character parts, which are occluded in the input image. We conclude that similar learning architectures may be used to model imitation behavior upon the observation of interaction events, because our model essentially learns to compactly and compositionally encode interaction events, and, meanwhile, learns to generate observed interaction events conceptually, by mimicry, or by blended versions of the two.

Method

Our generative model includes an image processing pathway, which is used both to classify the shown letter and to produce a compact code embedding $e_{bottom-up}$ for subsequent trajectory generation. A small subnetwork projects one-hot encoded letter classifcations onto the same compact code embedding space, yielding the top-down embedding $e_{top-down}$. The compact code, from either or a combination of both embeddings, then is passed through a long short-term memory recurrent neural network (LSTM, Hochreiter & Schmidhuber, 1997). After providing the details of our model, we specify the considered dataset and our performance evaluation approaches.

Model

The CNN-RNN model is depicted in Figure 1. It gets an image of a character as input and outputs the change in x and y position of the pen at every time step, thus generating a full trajectory. The first advancement in comparison to the simple LSTM model of Fabi et al. (2021b) is that the current model does not get a one-hot encoded vector, which encodes the character, but the image itself as an input. First, the input image is processed in a convolutional neural network with two convolutional layers with a kernel stride of 5x5 and 32 and 64 filters, respectively. Both convolutional layers are followed by a 2x2 pooling layer with stride 2. Dropout is applied with a probability of 0.5, the activations are flattened and fed into a dense layer of 100 units. Next, the two pathways are split up: While the first one transforms the output of the CNN directly into another fully-connected embedding layer $e_{bottom-up}$ with 100 neurons, the other pathway converts the output of the CNN via a softmax layer into a classification layer of size 26 (because of the 26 characters in the Latin alphabet). The layer's output is converted into a one-hot encoded vector, specifying the highest activation of the classification layer. The one hot code is transferred into the embedding $e_{top-down}$ with 100 neurons via a fully-connected layer. The activations of both layers are added neuron-wise with a fusion factor of w and $(1-w)$, such that either one of the layers or different combinations of them are processed further. This results in the embedding layer e_{total}, whose activities are then passed onto an LSTM (Hochreiter & Schmidhuber, 1997) layer with 100 units as a constant input. Starting from zero-initalized recurrent states, the LSTM layer then generates the output, that is, changes in x and y position. During training, the network was fed with pixel images of drawn letters and had as target output the class of the letter as well as the trajectory, which generated the image in the first place. To train via backpropagation, we calculated the cross-entropy loss between the correct and predicted classification vector plus the L2 loss between the original and the generated trajectory, which was weighted by a factor of 0.1 to adjust its range.

During training, the model learned to generate trajectories out of images of a subgroup of characters—in our case the first half of the Latin alphabet. The factor w, which determines the weighting of the bottom-up $e_{bottom-up}$ and top-down $e_{top-down}$ embedding adding up to e_{total}, is either fixed or randomly selected during training. In this way, the network has to be able to rely on both pathways, as well as combinations of them, essentially fostering the development of a common code for the embedding e, which is attempted to be generated by either pathway. All models were trained for 20 epochs, except for the model with only the top-down pathway which overfitted earlier and was therefore only trained for 10 epochs (cf. Figure 2). For the one-/few-shot generation, after training, the model was presented with one (in case of one-shot learning) or three (in case of few-shot learning) examples of a new character concept that had not been part of the training—in our case characters from the second half of the Latin alphabet. To let the network compositionally reassemble representations, which it had learned during training, for the generation of new trajectories, we allowed the three weight matrices into the classification layer and the two embedding layers to adapt for 2000 iterations. This is an adaptation of the one-shot inference mechanism applied to less complex models in previous work (Fabi, Otte, & Butz, 2021a; Fabi et al., 2021b). To avoid an unlearning of the training characters 'a' to 'm', 10 images of each character were presented at every 100th of the 2000 steps. All learning was performed using the L2 loss function at the trajectory output and the cross-entropy loss function in the classification layer. The Adam optimizer was used with standard parameters ($\eta = 0.0005$, $\beta_1 = 0.9$, $\beta_2 = 0.999$). Training batches had size 1.

Dataset

We used a dataset of 440 handwritten character trajectories of each character of the Latin alphabet recorded in our lab (Fabi et al., 2020). The characters were produced by experts (university students) of the alphabet using a dedicated pen on a touch-sensitive surface, leading to consistent and natural trajectories. The dataset provides natural variability from 10 different subjects, including script and print characters. For training, 80 % of the stimuli were randomly included in the training dataset and 20 % in the test set.

Hypotheses

We were not only interested in the overall performance in terms of prediction error of the bottom-up and top-down model, but also in the question which path combinations were most helpful for different tasks. Therefore, we looked at the performance of the two pathways separately as well as of an equal, biased,

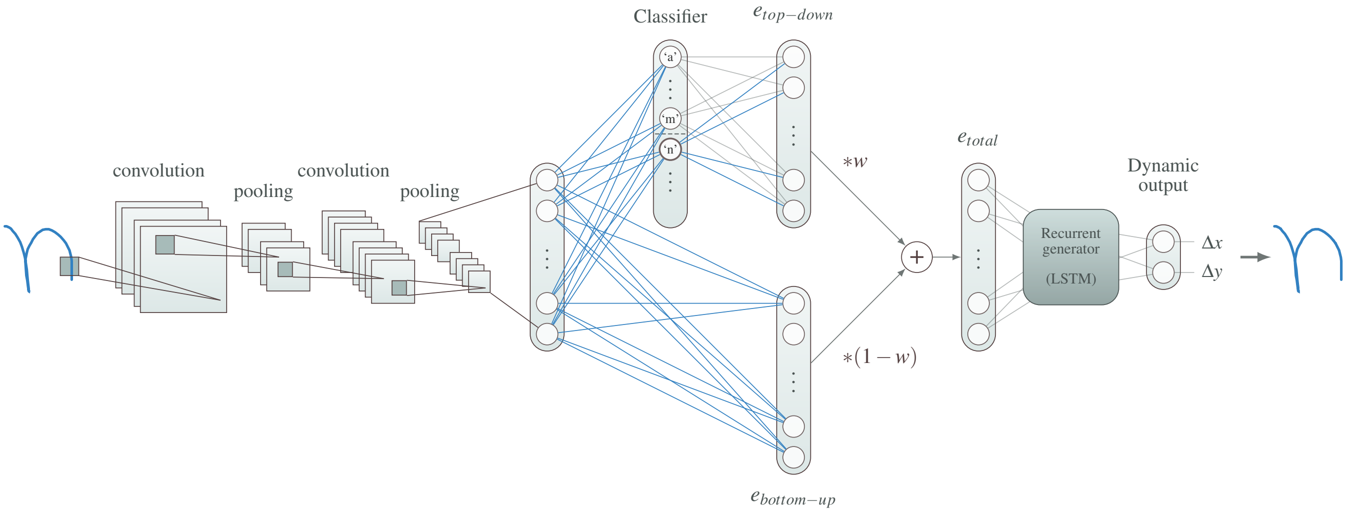

Figure 1: Illustration of the CNN-RNN model that takes a picture of a character as an input and outputs the change in x and y position for every timestep, thereby regenerating the trajectory. The upper part of the model shows the top-down pathway, which classifies the characters before generating the trajectory, whereas the lower part shows the more direct, bottom-up pathway. In order to flexibly use one or a combination of both pathways their embeddings are weighted by a factor w and $1-w$.

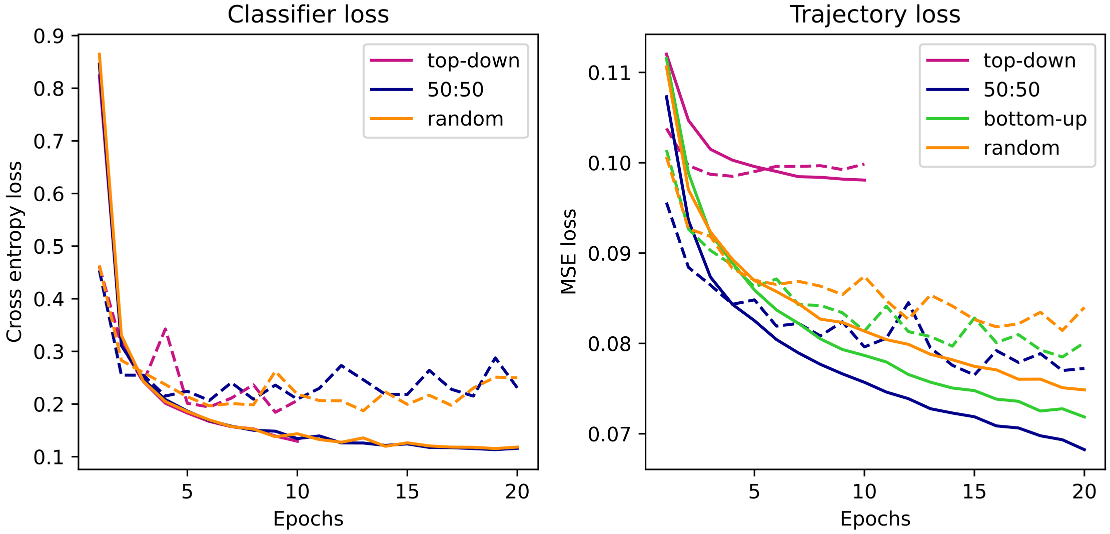

Figure 2: Classifier and trajectory loss for the train (solid line) and test (dashed line) sets for the models using only one pathway (top-down or bottom-up) or an equally weighed (50:50) or randomly chosen combination of both pathways.

and a random weighing of both pathways.

The first important question we wanted to answer was whether compositionality could also be incorporated in this complex system, that is, whether the model was able to extract components out of the basic training dataset of ‘a’ to ‘m’ in order to recombine them in a meaningful manner when confronted with new characters. We have previously shown that this is possible for a model that projects a one-hot encoded character into an embedding space, whose activities are then passed on as constant inputs to an LSTM module, which then generates the trajectory (Fabi et al., 2021b). This network corresponds to the top-down component used in our current architectures with weighting $w = 1$. Here, our goal was to additionally investigate whether compositional letter generation would also be possible with the current model, which generates trajectories out of pictures with the help of a top-down and a bottom-up pathway.

Next, we wanted to test whether the advancement of the model was in fact able to learn several instead of just one variant of characters in a few-shot manner. Here, we investigated whether the bottom-up pathway alone or a combination of both would lead to the best results. This task can be compared to learning new handwriting styles of a character, for which we hypothesize that humans use a combination of copying and drawing from memory.

Lastly, we hypothesized that even though we expected that adding the bottom-up pathway to the model should lead to advancements in performance, there might also exist tasks for which applying only the top-down pathway was more promising. In humans, classifying characters and writing them from memory might be most helpful when needing to refill characters that are partially occluded. This is why we introduced a new task where character trajectories had to be created from partially occluded images.

Results

One-shot Regeneration of New Characters

After the successful basic training on ‘a’ to ‘m’, we presented the model with one variant of characters ‘n’ to ‘z’. In order to learn those characters in an efficient manner, we wanted the network to recombine components like curves, straight strokes etc., which it had extracted previously in an unsupervised manner when trained on the first half of the alphabet. This is why we applied a variation of the previously employed one-shot inference mechanism (Fabi et al., 2021b): when presented with the second half of the Latin alphabet, instead of training the whole network, we kept everything the same except for the weight matrices into the classification layer and into the two embedding layers $e_{bottom-up}$ and $e_{top-down}$. Those were adapted by aiming at minimizing the standard loss, which we also used during training. In Figure 3, the model’s versions of the one-shot learned characters can be seen below the original characters. Shown are the results of the model using the two paths separately, as well as an equal or biased weighting of them. (Note that we used the same weighing during basic training and one-shot learning). Almost all characters are easily readable, with the equal combination of both paths leading to results that appear most similar to the original characters. Thus, the compositional structures, which develop in the latent states of the LSTM-based trajectory generator, can be efficiently exploited to learn new letters and to improve their similarity with novel letters—at least as long as they are drawn in a style that is comparable to the styles present during initial training (Fabi et al., 2021b). This is true for all pathways and combinations of those. We reported the results of models that had the same pathway combination during training and one-shot generation, but results looked similar when the pathway combination during training was random.

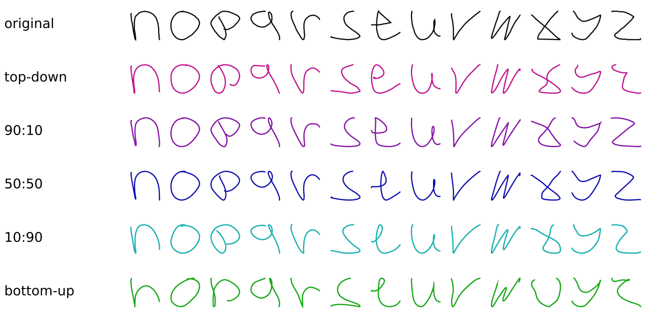

Figure 3: Original and efficiently generated trajectories of one variant of the characters of the second half of the Latin alphabet by reusing components extracted during training on the first half, when either using the top-down or bottom-up pathways separately, or a combination of them with equal (50:50) or biased weights (90:10 for a stronger focus on the top-down and 10:90 for a stronger focus on the bottom-up pathway). Results look most promising for the equal combination.

Generating Various Character Variants

In previous work, reusing previously extracted components was only possible when confronted with one variant of a character (Fabi et al., 2021b), because the input just encoded one particular character. Alternatively, the handwriting style had to be indicated as an additional input (Fabi et al., 2022). With the current model, the input could be any image of a character variant. To generate multiple character variants, the learning mechanism was expanded to few-shot learning, where the model received three versions per new character as learning input. As previously in one-shot learning, the characters presented in Figure 4 show the characters generated as a result of the few-shot learning process. Generating several variants was very good for the combination of the two pathways but not very well-suited for just a single pathway, leading to the conclusion that it is their interplay, which is really important in this architecture. While the top-down pathway secures that the correct character is drawn, the bottom-up pathway adds additional information about the specifics of this character variant. The three ‘n’s show nicely that the top-down path tries to generate one single version for all variants. The notable variation between the different character versions produced by the top-down pathway can be attributed to the different trajectory lengths of the original characters. As the architecture uses the same amount of time steps needed to generate the original characters in the character regeneration, the characters generated by the top-down path seem to differ from each other, when in fact they follow the same trajectory, which is simply cut off at different time steps. The bottom-up path, on the other hand, differentiates but is not able to grasp the details as good as the equal combination of both pathways, which regenerates all ‘n’s perfectly well.

When the character variants are too dissimilar, though, the model sometimes fails to generate appropriate output. For example, the second ‘q’ is written very differently from the others, which is why not all combinations but the equal combination of pathways is able to generate it in a readable manner. What is interesting here is that the top-down pathway generates a readable ‘q’, which looks very similar to the other ‘q’s. Just the difference in sequence length allows the extra curve. The letters would look the same if the underlying character sequences would be equally long.

Coming back to the equal combination of both pathways, even though there are limits, it is quite remarkable what the model is able to learn in this efficient manner, reusing previously extracted components. For example, the original second ’s’ looks very dissimilar from the others. Nevertheless, the model with the equal path combination is able to generate it very accurately. Also the ‘w’s or ‘y’s have very specific characteristics, which the combinatorial approach seems to catch. As can be seen in Figure 5, the model does not perform badly, even for a random path combination for each new picture stimulus, forcing it to deal with every possible combination of the two pathways.

Refilling Occluded Images

To further test model generalizability, we occluded one fourth of the picture stimuli during inference (either bottom or top). The task of the models, which were trained on the unoccluded versions of the stimuli in the one-shot learning process, was to refill the occluded portions. Our hypothesis was that this would work best if the network did not try to redraw the pictures with the bottom-up pathway, but if it identified the character and used the top-down pathway to generate it. The results for the different paths are presented in Figure 6. Indeed, the model seems to be especially good at this task when using the top-down pathway, indicating that the occlusions disrupt the information flow that attempts to convert image information into generative trajectory information directly.

Conclusion

The results of our analyses indicate that depending on the tasks at hand, the network performs best when using either one processing pathway or a combination of both. Our ANN architecture thus mimics how humans may learn to write,

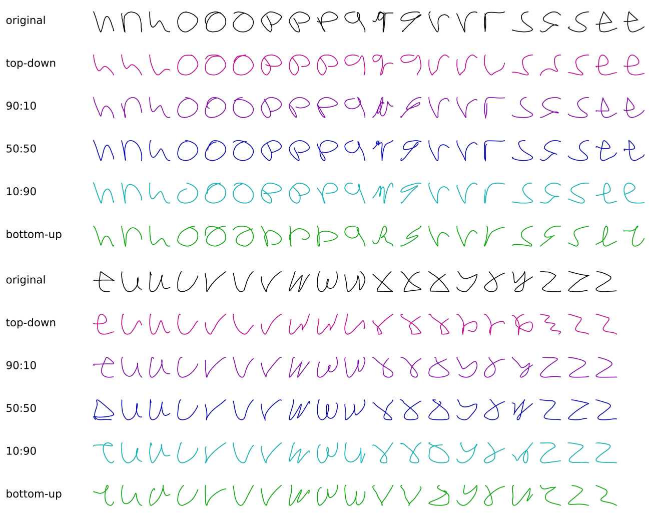

Figure 4: Original and efficiently generated trajectories of several variants of the characters of the second half of the Latin alphabet by reusing components extracted during training on the first half, when either using the top-down or bottom-up pathways separately, or a combination of them with equal (50:50) or biased (90:10 and 10:90) weights. The top-down pathway naturally generates only very similar looking variants, while the bottom-up pathway catches more fine-grained nuances, although in extreme cases of the input image, the output trajectory does not mimic the input letter type. Again, the combination of both pathways works best.

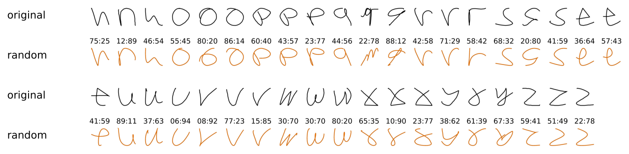

Figure 5: Original and efficiently generated trajectories of several variants of the characters of the second half of the Latin alphabet by reusing components extracted during training on the first half, using a random combination of the top-down and bottom-up pathways. Given above the generated characters are the randomly generated *w*-values determining the weight of the top-down pathway. The results confirm that smooth integrations of the two pathways are possible.

where in some situations (e.g., when imitating another person’s handwriting) we copy the depicted characters with some information about how the character is normally written, while in other situations (when refilling occluded images) we draw the identified character from memory.

In line with the Omniglot challenge (Lake, Salakhutdinov, & Tenenbaum, 2015; Lake et al., 2017, 2019), we have shown that our architecture is able to efficiently learn, most probably by recombining previously extracted components in a one-shot and few-shot manner even when starting from a pixel image (Fabi et al., 2021a). Thus, on top of our previous investigations

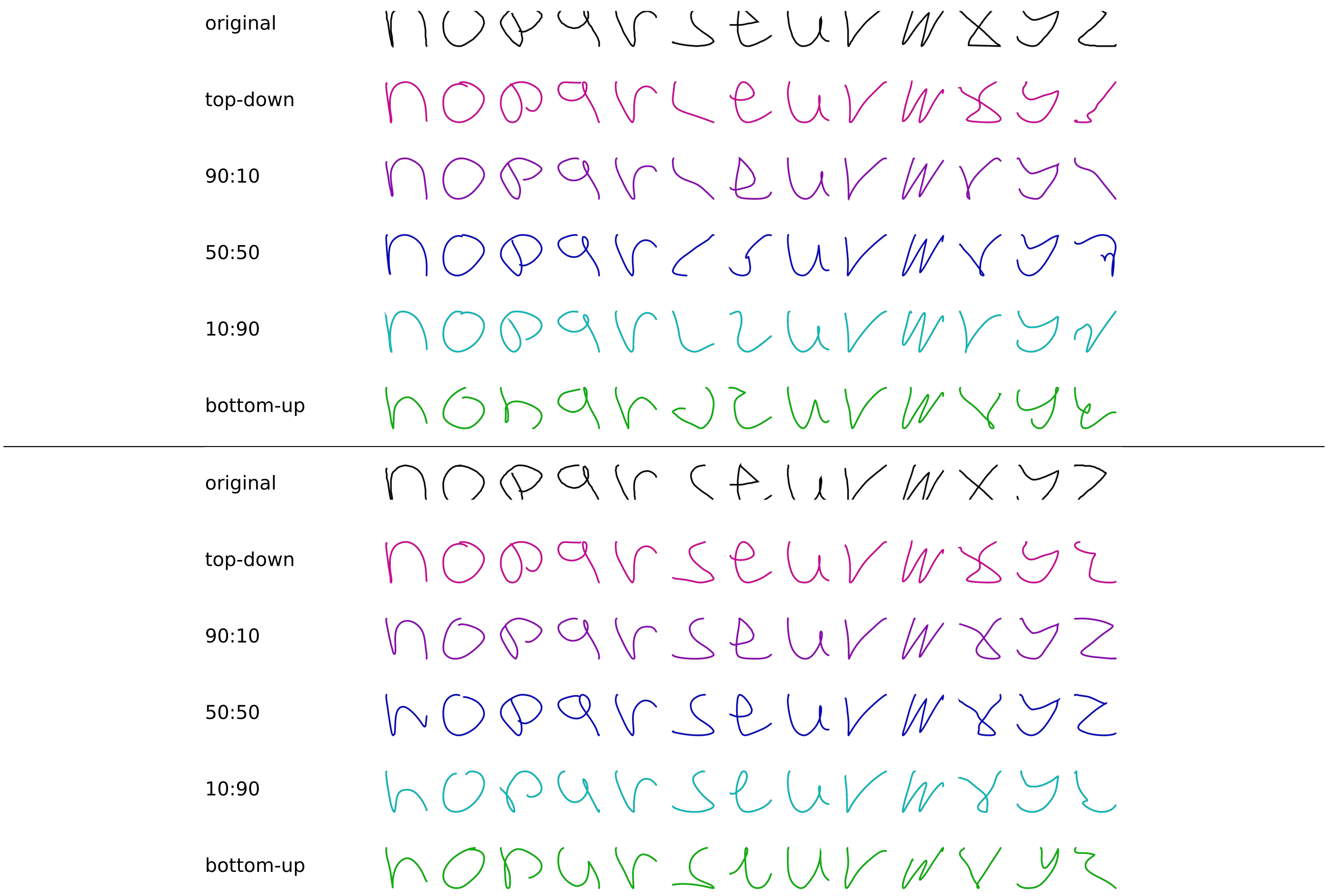

Figure 6: Occluded original input images (either top or bottom occluded) and refilled trajectories, when either using the top-down or bottom-up pathways separately, or a combination of them with equal (50:50) or biased (90:10 and 10:90) weights. The top-down path worked best for this task.

on compositional structures (Fabi et al., 2021b), we show that the development of internal, latent, compositional structures can also be elicited in more complex architectures, which combine convolutional neural networks that process images with a classification-based compact code for the generation of target trajectories. This leads the way to tapping into compositional structures in generative artificial neural network models on a larger scale, promising to improve their encoding efficiency even further.

While the combination of the bottom-up and top-down pathway worked well by weighing them task-suitably, future research should investigate whether the model can learn to select pathways flexibly and adaptively given a particular task at hand. In fact, the fusion weight w may depend on the current system intention, which would correspond to emphasizing to draw a letter from memory or to portray a given character image.

Overall, we hope that our work will be useful also in other tasks, where behavioral dynamics need to be generated top-down or interpreted, and possibly mimicked, bottom-up from sensory information about a particular interaction trajectory. The generation of a particular trajectory, as investigated herein, closely corresponds to all kinds of event-based interactions, which have been emphasized to be of paramount importance for the development of conceptual cognition and language competencies (Baldwin & Kosie, 2021; Butz, Achimova, Bilkey, & Knott, 2021; Elman & McRae, 2019; Franklin et al., 2020). When considering the development of imitation behavior in infants, our model may, for example, be used to selectively mimic the trajectory of an observed human behavior, or rather imitate the behavior conceptually in a goal-directed manner (Gergely, Bekkering, & Kiraly, 2002; Cuijpers, van Schie, Koppen, Erlhagen, & Bekkering, 2006). Along these lines, an important lesson from this model is that the fusion of top-down classification-based information and bottom-up sensory information can lead to the formation of common compositionally-embedded encodings. These encodings can then be used flexibly and adaptively to both infer compact interaction interpretations and selectively generate either conceptual goal-directed or stimulus mimicking trajectories as well as combinations thereof.

Acknowledgements

We thank Sebastian Otte for helpful discussions and Maximus Mutschler for maintaining the GPU cluster of the BMBF funded project Training Center for Machine Learning, on which the results were computed. This research was funded by the German Research Foundation (DFG) within Priority-Program SPP 2134 - project “Development of the agentive self” (BU 1335/11-1, EL 253/8-1). MB is part of the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645.

References

Baldwin, D. A., & Kosie, J. E. (2021). How does the mind render streaming experience as events? *Topics in Cognitive Science*, *13*, 79-105.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., … others (2018). Relational inductive biases, deep learning, and graph networks. *arXiv:1806.01261*.

Butz, M. V., Achimova, A., Bilkey, D., & Knott, A. (2021). Event-predictive cognition: A root for conceptual human thought. *Topics in Cognitive Science*, *13*, 10-24.

Butz, M. V., Bilkey, D., Humaidan, D., Knott, A., & Otte, S. (2019). Learning, planning, and control in a monolithic neural event inference architecture. *Neural Networks*, *117*, 135-144.

Cuijpers, R. H., van Schie, H. T., Koppen, M., Erlhagen, W., & Bekkering, H. (2006). Goals and means in action observation: a computational approach. *Neural Networks*, *19*(3), 311–322.

Elman, J. L., & McRae, K. (2019). A model of event knowledge. *Psychological Review*, *126*, 252–291.

Fabi, S., Otte, S., & Butz, M. (2021a). Compositionality as learning bias in generative rnns solves the omniglot challenge. In *Learning to Learn-Workshop at ICLR 2021*.

Fabi, S., Otte, S., & Butz, M. (2021b). Fostering compositionality in latent, generative encodings to solve the omniglot challenge. In I. Farkas, P. Masulli, S. Otte, & S. Wermter (Eds.), *Artificial neural networks and machine learning - ICANN* (pp. 525–536). Springer.

Fabi, S., Otte, S., Scholz, F., Wührer, J., Karlbauer, M., & Butz, M. (2022). Extending the omniglot challenge: Imitating handwriting styles on a new sequential dataset. *Under review*.

Fabi, S., Otte, S., Wiese, J. G., & Butz, M. V. (2020). Investigating efficient learning and compositionality in generative lstm networks. In I. Farkas, P. Masulli, & S. Wermter (Eds.), *Artificial neural networks and machine learning - ICANN* (pp. 143–154).

Franklin, N. T., Norman, K. A., Ranganath, C., Zacks, J. M., & Gershman, S. J. (2020). Structured event memory: A neuro-symbolic model of event cognition. *Psychological Review*, *127*(3), 327–361.

Gergely, G., Bekkering, H., & Kiraly, I. (2002). Developmental psychology: Rational imitation in preverbal infants. *Nature*, *415*(6873), 755–755.

Gopnik, A. (2019). AIs versus four-year-olds. In J. Brockman (Ed.), *Possible minds: Twenty-five ways of looking at AI.* New York: Penguin Press.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*, 1735–1780.

Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, *350*, 1332–1338.

Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2019). The omniglot challenge: A 3-year progress report. *Current Opinion in Behavioral Sciences*, *29*, 97–104.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, *40*, e253.

Otte, S., Schmitt, T., Friston, K., & Butz, M. V. (2017). Inferring adaptive goal-directed behavior within recurrent neural networks. *International Conference on Artificial Neural Networks (ICANN)*, 227-235.