Data driven algorithm design

Reinhard Heckel is a Rudolf Moessbauer assistant professor in the Department of Electrical and Computer Engineering (ECE) at the Technical University of Munich, and an adjunct assistant professor at Rice University, where he was an assistant professor in the ECE department from 2017–2019. Before that, he spent one and a half years as a postdoctoral researcher in the Department of Electrical Engineering and Computer Sciences at the University of California, Berkeley, and a year in the Cognitive Computing & Computational Sciences Department at IBM Research Zurich. He completed his PhD in electrical engineering in 2014 at ETH Zurich and was a visiting PhD student at the Statistics Department at Stanford University. Reinhard is working in the intersection of machine learning and signal/information processing with a current focus on deep networks for solving inverse problems, learning from few and noisy samples, and DNA data storage.

Reinhard Heckel

Data driven algorithm design

The past decade has seen remarkable advances in information processing, imaging, and sensing technologies by building data-driven solutions, even for problems that are traditionally solved without any data. For example, the newest generation of medical imaging technologies, such as computed tomography and magnetic resonance imaging, smartphones and now uses deep networks trained on example images to reconstruct an image from measurements. Those neural-network based algorithms yield significantly higher image quality and reconstruction speed than traditional algorithms that are not data-driven.

However, a common concern is that the resulting systems are data hungry, not well understood theoretically, and are sensitive to perturbations, ranging from biases towards the training data to sensitivity to adversarial worst-case perturbations. Our research addresses those concerns by developing robust algorithms, corresponding performance guarantees, and mathematical foundations for machine learning, information processing, and imaging systems.

In this article, we discuss our research on new algorithms and theory for solving image reconstruction problems with deep learning, and on systems that enable storing digital data on DNA.

1. Deep networks for inverse problems

An important problem in statistics, signal processing, and imaging is to reconstruct a signal or image from few and noisy measurements. Such signal and image reconstruction problems arise in imaging systems ranging from telescopes to medical imaging technologies such as magnetic resonance imaging and computed tomography.

For example, magnetic resonance imaging (MRI) is a non-invasive medical imaging technique that provides excellent soft-tissue

Resume:

The past decade has seen remarkable advances in imaging systems by training a neural network to reconstruct an image from measurements. For example, the newest generation of medical imaging technologies generate images with lower scan times and higher quality with those neural networks. However, the resulting systems are data hungry, not well understood theoretically, and are sensitive to perturbations, ranging from biases towards the training data to sensitivity to adversarial worst-case perturbations. Our research addresses those concerns by developing neural network based imaging algorithms that work with fewer data and are robust.
Due to its longevity and enormous information density, DNA is an attractive storage medium. However, practical constraints on reading and writing DNA require the data to be stored on many short sequences and introduce errors. We built the first robust DNA storage system that enables storing data on DNA reliably and developed coding theoretic and information theoretic foundations for future DNA storage systems.

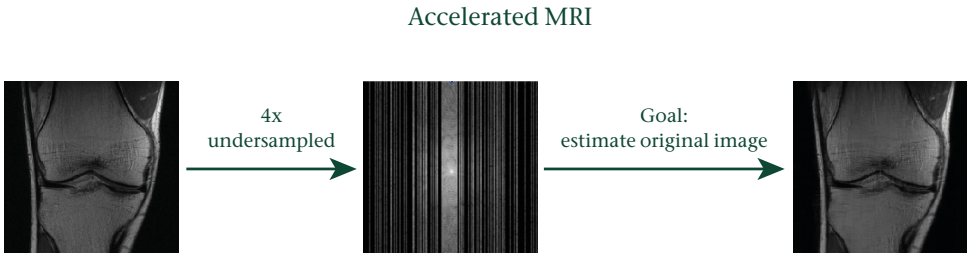


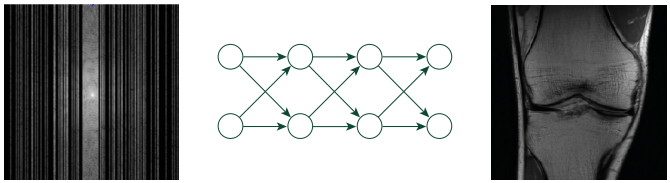
Figure 1:
The image reconstruction problem arising in magnetic resonance imaging. The image is undersampled by a factor of 4 which accelerates the scan time. The middle figure are the measurements in the frequency domain: each black line is a frequency that is not observed, and only ¼ of all frequencies are observed. The algorithmic problem is to reconstruct the original image from this undersampled measurement.

contrast, without using potentially harmful radiation. MRI is considered to be a safe diagnostic tool with the capability to reliably detect a wide range of diseases such as tumors, hemorrhage, and infections. However, the data acquisition in MRI is inherently slow, leading to time-consuming examinations. This makes MRI particularly difficult for patients who struggle to remain still for longer periods of time since even small movements increase the risk of image artifacts. To address this issue, MRI is usually accelerated by only collecting a few measurements *(see Figure 1)*. In order to reconstruct an image from such few measurements, we have to make prior assumptions on the image to be reconstructed.

Until recently, the best methods for reconstructing an image from few and noisy measurements were crafted by experts often based on the physics of the measurement process and without using data. Experts would handcraft intricate mathematical models for representing images and design algorithms that assemble an image from measurements using those models. However, it is very difficult to model images mathematically, and an algorithms performance is determined by the quality of models we use.

Today, deep neural networks trained on example signals perform best. This is testified by General Electric's newest computational tomography scanners and the iPhone using neural networks to produce high-quality images.

In a nutshell, those networks work as follows. We start from a large set of pairs of measurement and target image that we call



the training set. For example, if we train a neural network for the image reconstruction problem arising in accelerated magnetic resonance imaging, we first collect measurements generated by the MRI scanner along with a target image. As a second example, if we train a neural network for reconstructing a clean image from a noisy one, we generate pairs of noisy and clean image. For both examples, we then train a neural network, i.e., adjust it parameters so that the network maps the measurements or noisy images to the target images. *See Figure 2* for an illustration.

However, neural networks often rely on large amounts of data, which can be difficult and expensive to collect. In addition the dependency on data can introduce biases, for example, a neural network trained on patients from one hospital might perform significantly worse on patients from another hospital. Finally, we do not have the theoretical understanding of neural networks that we got accustomed too from classical imaging methods. This fuels concerns on whether we can fully trust the images generated by neural networks.

Figure 2:
Neural networks for image reconstruction are often trained end-to-end to reconstruct an image from measurements. That means the parameters of the network are adjusted so that the network maps the training measurements to the corresponding training images. If trained on sufficiently many such examples, the network can reconstruct an image from an under-sampled measurement.

1.1 Un-trained neural networks for image recovery

Neural networks for image recovery problems are typically convolutional neural networks and implicitly already incorporate many prior assumption on how natural images look like. Motivated by the fact that a major weakness of using standard neural networks for image recovery is their dependence on training data, we developed a convolutional neural network, called the deep

Original Image

Deep Decoder (almost perfect)

Traditional Method (visible artifacts)

Figure 3:
The deep decoder, an un-trained neural network, outperforms traditional image reconstruction algorithms for magnetic resonance imaging.

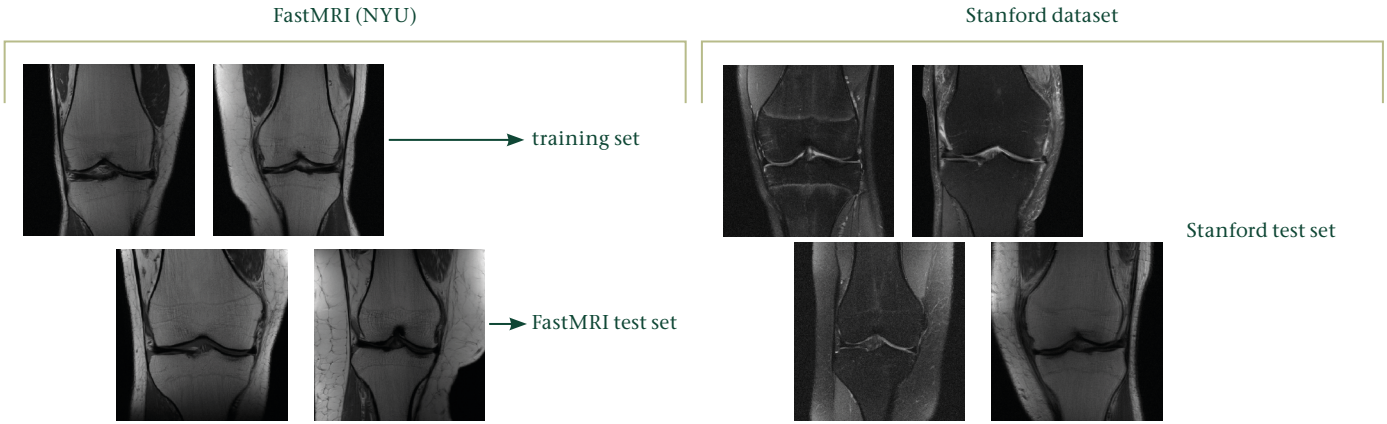
decoder, that enables image recovery without using any training data [1]. In a nutshell, in order to recover an image from a measurement, the deep decoder is fitted to a single measurement. The neural network itself acts as an image prior and only learns from the given measurement, but not from any extra training data.

For accelerated magnetic resonance imaging (MRI), the deep decoder significantly improves upon classical image recovery methods, and performs close to trained neural networks, but without the reliance on training data [4]. *See Figure 4* for an illustration. The deep decoder is broadly applicable to a variety of imaging problems. For example, beyond denoising and magnetic resonance imaging, in collaboration with Prof. Wallers group at Berkeley, we have shown that it enables high-quality phase microscopy [4].

” Deep neural networks outperform traditional image reconstruction algorithms both in image quality and computational cost. “

Reinhard Heckel

Unlike many competing methods, the deep decoder comes with rigorous mathematical performance guarantees: We developed mathematical denoising and signal recovery guarantees showing that the deep decoder can provably recovery a signal from few and noisy measurements [2,3]. Such performance guarantees are important as they build the trust necessary to apply neural network based methods in setup where accuracy and correctness is critical for performance, such as medical imaging, where an error in an image could lead to a misdiagnosis.



1.2 Towards robust imaging algorithms

While the deep decoder outperforms other traditional methods that rely on training data, it falls short of matching the image quality and low computational cost of neural networks trained end-to-end.

However, as mentioned previously, as data-driven algorithms are employed in medical and other systems, a concern among experts and the general public has grown that such systems might not be robust. A common concern is that an algorithm learned from data can be very sensitive to worst case perturbation. Neural network for classification have such sensitivities: an adversarial but small perturbation invisible for a human can fool an algorithm to missclassify a stop sign for not-a-stop-sign for example.

Another concern is a bias towards the training data. For example, a super-resolution algorithm trained to generate high-resolution images from low-resolution one has recently been shown to generate a face of a white male from a low-resolution image of Barack Obama, the first black president of the US.

We have recently shown that distribution shifts such as training on data from one hospital and testing in another hospital, or training on knee images and applying the method on brain images leads to a significant performance degradation of current neural networks for image recovery problems. For example, a neural network trained for reconstructing knees in accelerated magnetic resonance imaging (MRI) does not reconstruct brains well, even though the same network trained on brains reconstructs brains perfectly well. *See Figure 5* for an illustration.

Figure 4:
A neural network trained on a training set and tested on test set from a common source performs much better than a neural network applied to entirely different data, such as the Stanford set above. This is called a distribution shift. We developed neural network based methods that also perform well under such extremely common distribution shifts.

Thus there is a distribution shift performance gap for a given neural network, defined as the difference in performance when training on a distribution P (for example knees) and training on another distribution Q (for example brains), and evaluating both models on Q. It is critical to develop methods to overcome this distribution shift performance gap, since method are routinely applied to different data than they are trained on.

” Deep neural networks do not need to be data hungry or sensitive to perturbations, to some extent we can trade off computation and amount of data: By investing more compute we can perform well even if we’re given little training data. “

Reinhard Heckel

We proposed a domain adaptation method for deep learning based imaging methods that relies on self-supervision during training paired with test-time-training at inference. We show that for three natural distribution shifts, this method closes the distribution shift performance gap for state-of-the-art architectures for accelerated MRI.

1.3 Outlook

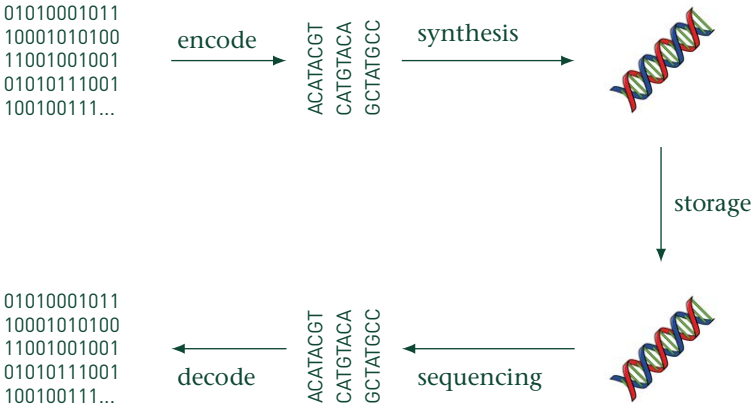
Algorithms in computer science and engineering are traditionally designed without learning. However, as the results in computational imaging mentioned above testify, a data-driven approach to algorithm design can lead to significantly better performance by fine-tuning algorithms to properties of the data they are applied to. We are very excited about the potential of data-driven algorithm design and we and others are currently developing new data-driven algorithms beyond imaging. For example, we are exploring data-driven algorithms in communication systems and genomics.

DNA data storage

Over the centuries, data has been stored on a variety of media: from paintings on rocks to books to digital storage media such as punch cards, compact discs, hard drives and magnetic tapes. In the last century, digital storage media has become ubiquitous, and data of any kind is almost exclusively recorded digitally as a sequence of zeros and ones.

The amount of data we generate and store each year has grown exponentially over the last few decades and is in the zettabyte range this year. A zettabyte is a sequence of zeros and ones of length 8*10^21. To illustrate how huge this amount of data is, let's imagine that we were writing a zettabyte into average-sized books. If we lay out these books on the earth, they would cover the earth five times. This illustrates that books are not suitable for storing large amounts of data simply because the required data density is so high. By data density we mean the amount of data per volume.

Even though the amount of data we store is constantly growing and the digital storage technologies on which we save the data are constantly evolving, the goal of each of these storage technologies has remained



the same: to store data for the long term and make it easily accessible.

It is perhaps surprising, then, that digital data typically does not have a particularly long shelf life. A typical digital camera does not store photos for more than a year. Hard drives, where the majority of the world's digital data is stored, only last about three to twenty years.

This motivates an alternative to conventional storage technologies such as hard drives: data storage on DNA. DNA is nature's storage medium. There, the instructions for the development, function, and growth of all known organisms are stored as a long chain of molecular building blocks called nucleotides, abbreviated A, C, G, and T. The DNA of a very simple organism is a chain of 200,000 such building blocks, and that of humans consists of 3 billion nucleotides.

There are two reasons why DNA is interesting as a digital storage technology: DNA has an extremely high information density and is very durable. In theory, one can store a zettabyte in just 5 grams of DNA. If the same amount of data were kept in books, those books would cover the earth five times over. In practice, one comes very close to this information density. The second major benefit of DNA is longevity. DNA can be kept for a very long time when it is dry and cool: Nature provides the proof, DNA from a million year old horse bone has recently been successfully sequenced.

In theory, storing digital data on DNA is very easy. Digital data such as text, image,

Figure 5:
Illustration of a DNA storage system: An algorithm translates a file to a set of DNA sequences. Those are synthesized and stored. At reading, the DNA is first sequenced and then an decoding algorithm reconstructs the information from the reads.

video and any other file is made up of a sequence of numbers 0 and 1. Similarly, DNA is made up of a sequence of nucleotides A, C, G, and T. A digital file consisting of bits can easily be translated into a sequence of nucleotides using the translation 00 – A, 01 – C, 10 – G, and 11 – T. For example, the digital sequence 01 00 11 becomes the DNA sequence CAT. In principle, data can be stored so easily on DNA.

In practice, however, it is very difficult to store data on DNA because it is complicated to write (synthesize) and read (sequence) DNA. One reason for this is that DNA is extremely small: a nucleotide, the building block of DNA, consists of less than 40 atoms and is around one nanometer in size. DNA sequencing technologies which can read short sequences of DNA are relatively well developed since there are plenty of applications in medicine, biology, and chemistry that have driven this development.

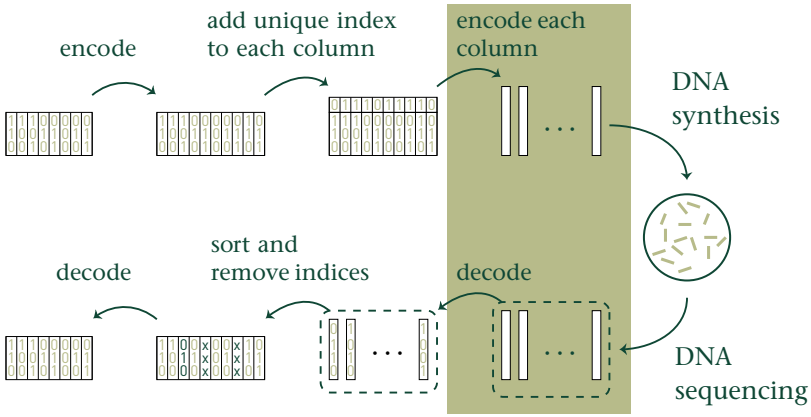
"I'm very excited about the potential of data-driven algorithm design and we and others are currently developing new data-driven algorithms beyond imaging."
Reinhard Heckel

Unlike DNA sequencing, DNA synthesis, the writing of DNA, is significantly less developed simply because there are fewer commercial applications for it at the moment. Since the 1980s, various DNA synthesis technologies have been developed that generate DNA strands by sequentially joining nucleotides using various techniques. These technologies make it possible to generate large numbers of sequences simultaneously, but as the sequences get longer, it becomes more difficult and error-prone to add new nucleotides. For this reason, current methods can only produce relatively short DNA strands of 100-300 nucleotides in length.

So we can write and read DNA and thus in principle store data. However, due to the existing synthesis and sequencing technologies it is only possible to write relatively short sequences consisting of about 100 nucleotides. Also, DNA synthesis and DNA sequencing technologies make many errors in writing and reading; it is not uncommon for whole sequences to go unwritten. In addition, sequences decay during storage which leads to breakups of sequences and to mutations of bases.

For data storage, however, it is absolutely necessary that we can read the written data without errors. To make this possible, we use two methods. First, we physically protect the DNA to minimize errors. That's similar to protecting a hard drive with a case, only on a much smaller scale, by embedding the DNA segments in nanometer-sized balls. Second, we protect the data algorithmically by adding extra information to the written data. That's a technique we use every day in every data transmission and storage technology, such as our computers, the Internet, and cell phones. It's called channel coding.

How is it possible to save data error-free using algorithms? To better understand this, it



helps to think about the language. It is easy for us humans to understand a sentence even if the sentence has many spelling mistakes. This is only possible because our language is very redundant, which means it contains more information than is necessary for understanding. Similarly, an algorithm can artificially add information, and then when mistakes happen, the algorithm can detect and correct the mistakes. Thus, the original data is restored perfectly from a very corrupt version of the data.

In 2015, we developed error-correcting codes specifically for DNA storage [9]. Those codes take the information, add redundancy, and translate it to a set of DNA sequences. Those DNA sequences are then synthesized and stored in DNA. We then read a noisy set of sequences, which contains many of the original sequences many times but with multiple errors (e.g., we read ACT instead of AAT) and some sequence are never read (5–50% of all sequences for example). In addition, the sequences are unordered. Our algorithms can then reconstruct the original information from those sequences perfectly, making use of the information we build in.

In 2015, in collaboration with Chemistry Professor Robert Grass at ETH, we built the first robust DNA storage system with those error-correcting codes we developed and with a technology for encapsulating DNA that Professor Grass developed. With experiments simulating accelerated aging, we demonstrated [9], for the first time, that data stored in this way on DNA will last for at least 1000 years – far longer than with conventional storage technologies.

Our coding scheme uses outer to protect from the loss of sequences and an index to retrieve the order of the sequences. It is not clear a priori whether this coding scheme is optimal, i.e., whether it uses the minimum

Figure 6:
An illustration of our error-correcting scheme. We first split the information into blocks, encode them with an outer code, and add an index to each symbol (column in the figure) of the outer code. We then encode each column with an inner code. From sequencing, we get a set of noisy reads, and some of the original sequences are never read. Our algorithm first decodes the inner code which corrects errors within sequences. Then the algorithm sorts the sequences by indices. Now a fraction of the sequences are lost, but our outer decoder can recover those lost sequences and perfectly reconstruct the information.

Figure 7:
The first commercial application of DNA data storage. We stored an album on a Million DNA sequences.

amount of redundancy given a level of errors. We developed information theoretic bounds that rigorously established that, indeed, our scheme is information theoretically optimal [10]. Subsequently, we also studied more involved error models where errors are also introduced on the sequence level, and we also established information theoretic bounds and found optimal coding schemes for this regime.

Which coding scheme to use depends on the error statistics, and it is therefore critical to understand the error statistics as a function of the various technologies (sequencing, synthesis, etc) and as a function of storage conditions. We empirically studied and characterized those error statistics [11], and together with the error correcting schemes, our information theoretic understanding, and the empirical characterization we now have many tools in place to build well performing DNA storage systems [14].

> *”DNA is a very promising digital storage medium for its longevity and information density.“*
>
> *Reinhard Heckel*

In 2018 and 2020, we used our algorithms in collaboration with Prof Grass's group for the first commercial application of DNA storage. Specifically, we storing an album from the British Rock band Massive Attack music on DNA and mixed this into spray paint. A musician in the band used it to create graffiti, and each tiny splatter of paint in this artwork contains enough DNA to accurately reconstruct the music. That's an application hardly imaginable implementable with a harddrive or another traditional storage media. Subsequently we stored an episode of

the TV Series 'biohackers' commercially for Netflix on DNA.

Does this mean that our digital data will soon be stored on DNA? Probably not, as storing data on DNA is currently very expensive and much slower than storing data on a hard disk. At the moment this is primarily due to the writing of the DNA (DNA synthesis) and secondarily to the reading (DNA sequencing). Of course, DNA synthesis and sequencing technologies will advance rapidly and become cheaper over the next few years, and it is quite normal for a new technology to be very expensive. For comparison: The first commercial hard disk, the IBM 350 storage unit from 1956, could only store 3.75 MB and was rented for $3500 per month. At $500 per MB we are definitely cheaper for eternity, but of course we are still far away from storing data on DNA on a large scale.

A possible commercial application for DNA as data storage is the archiving of very valuable data that we want to keep for a very long time, like the results of scientists, art and culture, but also increasingly elements like computer code that make our world work. GitHub, a company that manages public computer code on the Internet, has impressively demonstrated this by launching the Arctic Code Vault program in 2020: a program in Antarctica where computer code is stored under extremely stable conditions for future generations.

" Digital information can only be stored on a set of very short DNA sequences, and since many sequences are lost and the others contain errors, storing information on DNA is a difficult algorithmic problem. "

Reinhard Heckel

It cannot currently be predicted if and when DNA will become commercially established as a digital mass storage medium. However, this molecule certainly has a role in information technology systems and thus outside of its biological origin. Products are already being marked with short synthetic DNA codes, and we have shown that DNA, in addition to storing and transmitting information, can also be used to generate random numbers that are necessary for cryptography and to carry out complex calculations. DNA might thus have a bright future in next generation computer systems.

References

[1] R. Heckel and P. Hand, “Deep decoder: Concise image representations from untrained non-convolutional networks”, ICLR, 2019.

[2] R. Heckel and M. Soltanolkotabi, “Denoising and regularization via exploiting the structural bias of convolutional generators”, ICLR, 2020.

[3] R. Heckel and M. Soltanolkotabi, “Compressive sensing with untrained neural networks: Gradient descent finds the smoothest approximation”, ICML, 2020.

[4] M. Darastani, R. Heckel, “Accelerated MRI with untrained neural networks”, IEEE Transactions on Computational Imaging, 2021.

[5] M. Darastani, A. Chaudhari, R. Heckel, “Measuring robustness in deep learning based compressive sensing”, ICML 2021 (long talk, top 3%).

[6] E. Bostan, R. Heckel, M. Chen, M. Kellman, L. Waller “Deep phase decoder: Self-calibrating phase microscopy with an untrained deep neural network”, Optica, 2020.

[7] R. Heckel, N. B. Shah, K. Ramchandran, and M. J. Wainwright “Active ranking from pairwise compar- isons and when parametric assumptions don’t help,” Annals of Statistics, 2018.

[9] R. Grass, R. Heckel, M. Puddu, D. Paunescu, and W. J. Stark, “Robust chemical preservation of digital information on DNA in silica with error-correcting codes,” Angewandte Chemie International Edition, 2015, featured in Nature as Research Highlight, press coverage by BBC Future, CNN.

[10] R. Heckel, I. Shomorony, K. Ramchandran, and D. Tse, “Fundamental limits of DNA storage systems,” ISIT, 2017.

[11] R.Heckel, G.Mikutis, and R.N.Grass,“A characterization of the DNA data storage channel”, Scientific Reports, 2019.

[13] R. Heckel and F. F. Yilmaz, “Early stopping in deep networks: Double descent and how to eliminate it”, ICLR, 2021.

[14] P. Antkowiak, J. Lietard, M. Z. Darestani, M. Somoza, W. Stark, R. Heckel*, R. Grass*, “Low cost DNA data storage using photolithographic synthesis and advanced information reconstruction and error correction”, Nature Communications, 2020, (*=corresponding authors).