

Deep learning enabled design of complex transmission matrices for universal optical components

Nicholas J. Dinsdale,^{1,2} Peter R. Wiecha,^{2,3,*} Matthew Delaney,^{1,2} Jamie Reynolds,¹ Martin Ebert,¹ Ioannis Zeimpekis,¹ David J. Thomson,¹ Graham T. Reed,¹ Philippe Lalanne,⁴ Kevin Vynck,⁴ and Otto L. Muskens^{2,†}

¹*Optoelectronics Research Centre, Faculty of Engineering and Physical Sciences,
University of Southampton, SO17 1BJ Southampton, UK*

²*Physics and Astronomy, Faculty of Engineering and Physical Sciences,
University of Southampton, SO17 1BJ Southampton, UK*

³*LAAS, Université de Toulouse, CNRS, Toulouse, France*

⁴*LP2N, CNRS - Institut d'Optique Graduate School - Univ. Bordeaux, F-33400 Talence, France*

Recent breakthroughs in photonics-based quantum, neuromorphic and analogue processing have pointed out the need for new schemes for fully programmable nanophotonic devices. Universal optical elements based on interferometer meshes are large compared to the limited chip real estate, restricting the scalability of the approach. Here, we propose an ultracompact platform for low-loss programmable elements using the complex transmission matrix of a multi-port multimode waveguide. Our approach allows the design of arbitrary transmission matrices using patterns of weakly scattering perturbations, which is successfully achieved by means of a deep learning inverse network. The demonstrated platform allows full control over both the intensity and phase of all outputs in a 3x3 multiport device using a footprint of $33 \times 6 \mu\text{m}^2$ and for typical perturbations achievable in experiments.

keywords: multimode interference devices, deep learning, inverse design, complex transmission matrices, light routing, analog optical computing

I. INTRODUCTION

Programmable photonic integrated circuits have been introduced as a new paradigm for the datacommunications domain, to circumvent the limitations of application-specific fabrication that is required for even the smallest changes in design¹. Reconfigurable components are finding immediate applications in post-fabrication trimming², tunable elements for microwave photonics³, and optical switch matrices⁴. However, emerging areas are on the horizon and quantum and neuromorphic computing have been recently seeing breakthrough developments⁵⁻⁸. The conventional engineering toolkit for such reconfigurable photonic networks consists of interferometer meshes and directional couplers, which have been successfully applied to achieve universal optical elements^{9,10}. Ultimately, an approach based on cascading of many single-mode devices to obtain a multi-mode logical circuit will face the limited real estate available on a silicon chip, limiting the scalability of this approach.

Generally, the input-output relationship of linear optical elements is fully described by its transmission matrix, and multiple scattering allows strong mixing of input and output ports¹¹. Here, each output port contains the full information of all inputs in a scrambled manner, with for example applications in compressive sensing. Control over the transmission can be achieved via the shaping of the phase and amplitudes of the input degrees of freedom, thereby allowing the descrambling of the information from a multiple scattering pattern¹². Alternatively one can control the transmission matrix of the medium itself by carefully positioning the individual scatterers, however, this is a problem which quickly diverges with the number of scattering sites. Nevertheless, scattering devices that are based on complex nanostructuring of a photonic waveguide have been proposed that can be optimized for a particular task¹³⁻¹⁶. In these devices, strong multiple scattering of photons is used to achieve ultracompact de-

vice footprints. Similar complex structured devices are currently being considered for emerging applications in analogue computing^{17,18}. The basic principle here is that the complex medium allows the modeling of a specific mathematical function that performs a linear operation on an input signal.

As an intermediate solution between strong multiple scattering designs and weaker diffractive optical elements, the application of patterns of small perturbations to a multimode interference device was recently shown to result in a programmable integrated spatial light modulator on silicon¹⁹. The weak perturbation limit is very different from the strong scattering-based complex devices and is more related to, for example, branched-flow scattering systems²⁰. A particular advantage is that weak perturbations allow controlling the flow of light with very high overall throughput and minimal reflection losses.

Here, we demonstrate that weakly perturbed multimode waveguides in fact allow full control over the transmission matrix, enabling the design of universal optical components. We successfully address the challenge of inversely designing the scattering medium by using a neural network approach. Deep learning^{21,22} has proven in the last few years to offer powerful tools for data processing and solving of problems with tremendous complexity, and has ever since been considered of highest relevance for photonics by the scientific community. Various impressive applications have been reported in (nano-) photonics like phase recovery in optical microscopy,²³ optical characterization and classification of nanometer-scale specimen²⁴⁻²⁶ or the real-time interpretation of light scattering through complex media²⁷⁻²⁹.

Particular research interest has been put into what is usually called *inverse design*, the design of a device offering a specific, pre-defined functionality. There have been numerous demonstrations of etched silicon perturbation patterns for wavelength and modal multiplexing^{15,16}, polarization beam-splitting³⁰, tight radius waveguide bending³¹ and free-space-

to-waveguide coupling³². Unfortunately, even for the most simple case of a binary perturbation pattern the problem contains 2^P possible solutions, where P is the number of possible perturbation positions. The search for appropriate patterns is therefore usually based on topological optimization, evolutionary algorithms or other computationally very expensive heuristics^{15,16,19,33–39}.

A deep artificial neural network (ANN) can in principle be trained on the solution of inverse problems and can, in particular, learn to inverse design nano-photonic devices^{40–44}. Transmission matrix design using sub-wavelength refractive index perturbations represents an ideal challenge for this novel ANN approach, owing to the complexity of the patterns required and the lack of a fast numerical solution for pattern optimization. In this work we leverage the power of full-wave Maxwell's equations solvers to generate a library of solutions for random perturbation patterns within multimode interference devices (MMIs) with different port configurations, and train a neural network to be able to predict perturbation patterns, in real time, to create an arbitrary transmission matrix. We demonstrate that our ANN approach allows not only to tailor incoherent transmitted intensities, but also can be used to gain independent control over the phase in each transmission channel. The full control over matrix elements enables rational design of universal optical elements and multi-port unitary operators.

II. PROBLEM, MODEL AND NEURAL NETWORK ARCHITECTURE

A. MMI device model

As a demonstration of capabilities, we consider M -input by N -output MMI rib waveguides with lateral dimensions of $33 \times 6 \mu\text{m}^2$ obtained by etching 120 nm into a 220 nm thick silicon layer of an silicon-on-insulator (SOI) wafer. The device is patterned by non-overlapping square perturbations of $750 \times 750 \text{ nm}^2$ in size, resulting in a 8×44 grid of possible positions. Consequently, our geometry allows 2^{352} (around 10^{106}) possible different patterns, a number which renders a systematic search for specific MMI designs impossible. The input and output waveguides are single mode with a width of 500 nm and taper up to $1 \mu\text{m}$ wide at the MMI region boundary over a length of $10 \mu\text{m}$, thereby allowing the adiabatic expansion of the fundamental TE mode and reducing transmission losses⁴⁵. In our studies we focus on two cases, respectively corresponding to 1×2 and 3×3 multi-port MMIs. In each case, the waveguides are centered with respect to the MMI and are separated by $3 \mu\text{m}$ (in case of 2 ports), respectively $2.15 \mu\text{m}$ (in case of 3 ports).

B. Training data generation for arbitrary transmission states

A systematic evaluation of all 2^{352} possible patterns is impossible. However, only a very sparse subset of these patterns provides a high overall throughput as light is usually scattered out of the device and does not arrive at the output waveguides.

Consequently, a network trained on exclusively random perturbation patterns will not manage to generalize to those desired patterns which result in high throughput solutions with specific splitting ratios. Therefore, in order to improve the quality of the solutions, it is critical that the network is trained on a set of solutions with reasonably high throughput. This is why prior to the training of the actual ANN we first need to develop a method to generate a training set of high-performance MMIs, which still contain a sufficient amount of randomness in their perturbation layouts.

The overall scheme of our data generation process is presented in Fig. 1a. The generation of an initial set of patterns with high throughput, “step 0”, is done by maximizing the following fitness function

$$f = 1 - \left[\sum_{m=1}^M \sum_{n=1}^N (T_{m,n}^{\text{tar}} - T_{m,n})^2 \right]^{0.5}, \quad (1)$$

where $T_{m,n}^{\text{tar}}$ and $T_{m,n}$ are the target and modelled total transmittance for the corresponding coupling between the m^{th} input and n^{th} output ports. Randomized splitting ratios are chosen as target values, using a normalization scheme for the total transmittance as explained in the supporting information section VI A. During this iterative process, perturbations are added successively one-by-one in a randomized order of positions for the grid. Each pattern is evaluated by a full-field numerical Maxwell solver. If the fitness function is improved once a new perturbation was added, this perturbation is kept. Perturbations that do not improve the throughput are discarded. The process is repeated until one of following stop criteria is met: either the fitness function f cannot be further improved, or the maximum allowed number of 50 accepted perturbations is reached. Practically, on average the fitness cannot be further improved after around 30 accepted perturbations. Figure S1 in the supporting information illustrates an example of pattern generation for the optimization of coupling towards a single output of a 1×2 MMI.

In principle any Maxwell solver can be used for the evaluation of the fitness function f . For the initial pattern generation (step 0), we use an in-house implementation of the aperiodic Fourier Modal Method (a-FMM)^{46,47}, relying on a supercell method and perfectly matched layers to compute the modes of the system, and a scattering matrix formalism to describe mode coupling. The method has previously been shown to provide numerical predictions with high accuracy and fast convergence⁴⁸. It has also been demonstrated to be well-suited to the study of multi-port MMI devices containing perturbations^{19,47}, thanks to its very fast runtime and direct access to the full set of coupling coefficients between all input and output waveguide modes. Simulations are performed in 2D using the effective index method⁴⁹.

Subsequently, in step 1 of Fig. 1a we simulate all patterns using finite-difference time-domain (FDTD) simulations (Lumerical FDTD), using the same parameters as with the a-FMM approach. We change numerical method to be closer to the experimental situation as the tapered input and output waveguides cannot be efficiently modelled with the a-FMM. The electric field intensity profiles across the single-mode out-

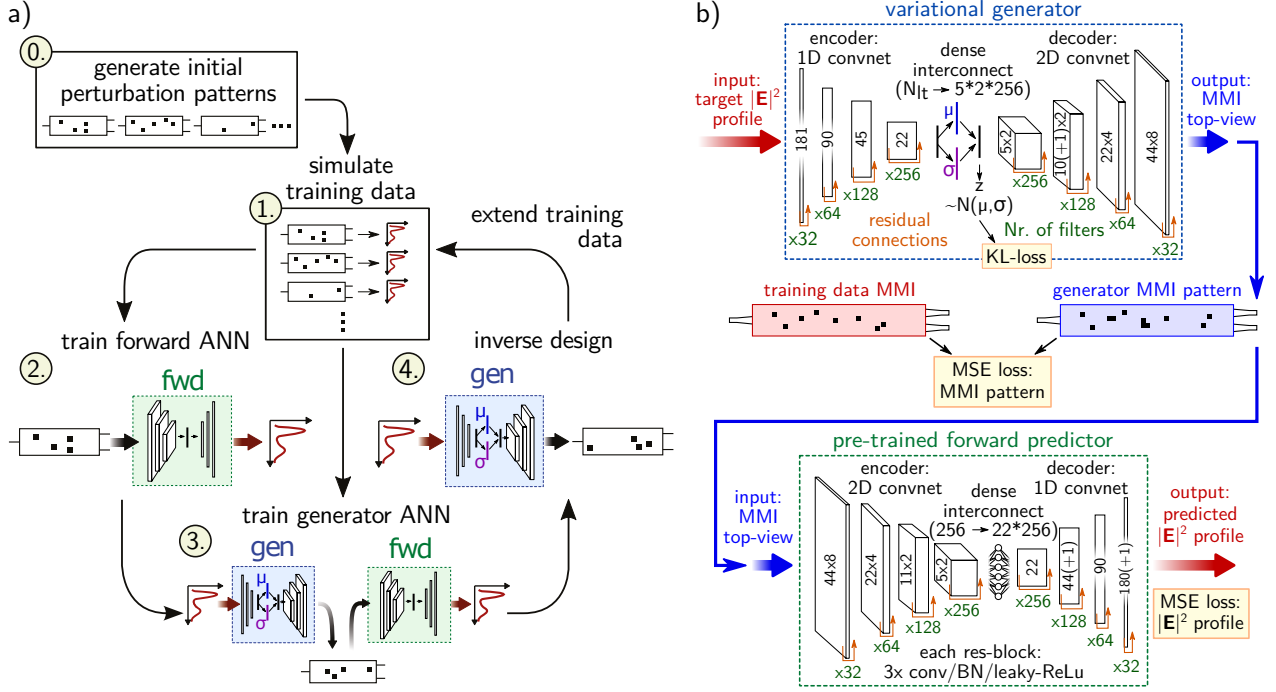


Figure 1. (a) scheme of the iterative data generation procedure. An initial dataset is generated, simulated and subsequently used for ANN-training. First the forward network is trained using only the intensity-profile loss. The forward network’s weights are then fixed and the generator network is trained. The trained generator ANN is then used to inverse-design new patterns, which are simulated and added to the training data. (b) detailed overview of the full inverse-design ANN model, consisting of variational generator (blue box) and pre-trained forward predictor network (green box). The full model is trained using the sum of three losses: MMI-pattern loss, $|E|^2$ -loss and KL-loss with weightings 1 : 50 : 0.2, respectively.

put waveguides are calculated at a distance of $20\mu\text{m}$ from the MMI end surface to estimate the out-coupling into the single-mode waveguide sufficiently far away from the taper. For each pattern, the transmittance profiles are simulated with an incoming fundamental TE mode into each input port (at $\lambda_0 = 1550\text{ nm}$). Finally, the size of the dataset is doubled by exploiting the symmetry of our systems about the propagation axis. All subsequent simulations are made with FDTD using the same spatial and temporal resolution.

The initial datasets (for both the 1×2 and 3×3 MMI cases) comprise of a total of 2500 of MMI-“families”, where each family corresponds to the ensemble of accepted iterations in an optimization cycle. After exploiting the mirror symmetry, the generated datasets consist of around 150000 patterns each.

C. Tandem neural network layout

We here use a network model based on a combination of two encoder-decoder type ANNs. Each network is composed of so-called “residual” down- and up-sampling convolutional blocks. Such “ResNets” allow to maintain efficient training convergence of very deep network layouts⁵⁰.

The first part for our ANN model is a forward network, which acts as a “physics predictor”. A detailed overview of the predictor network structure is given by the green box in

figure 1b. It takes as input a 2D perturbation pattern and returns the predicted 1D output intensity profile. During the forward training (step 2 of Fig. 1a), the ANN-predicted intensity-profiles are compared to the numerically simulated MMI output, calculating a mean-square-error (MSE) loss (“ $|E|^2$ profile” loss). By minimizing this loss (via the ADAM minimizer⁵¹), the ANN learns to predict the physical response of the MMI. We use 90% of the dataset for the actual training and keep 10% for validation, which is important to confirm the generalization of the network. Trained on this initial training data, the forward network achieves reconstruction of the field intensity at the output ports with a $<5\%$ average error. Once trained, the forward-ANN parameters are fixed during the subsequent inverse-network training step.

In step 3 of figure 1a, we introduce the second component of our model, the generator ANN, network details are given by the blue box in Fig. 1b. The generator takes the inverse design target as input, which is here the intensity profile across the output waveguides of the MMI. In the case of multiple input channels, the ANN takes a separate intensity target for each input. The generator ANN returns an MMI perturbation pattern. The generated pattern together with the training-data pattern yield a mean-square error (MSE) loss used for training of the network (the “MMI pattern”-loss in figure 1b).

It has been shown that the pattern loss alone is not sufficient for a stable training of “many-to-one” problem solver ANNs

such as our inverse design network⁵². Therefore, in addition to the pattern loss, the pre-trained forward ANN is used to circumvent the “many-to-one” problem, which is the fact that many different patterns may lead to an identical (or very similar) transmission through the MMI and hence the solution is non-unique. To avoid that the network gets confused by different solutions to the same problem, the generated MMI patterns are fed back into the pre-trained forward ANN. Its predicted transmittance is compared to the training data, yielding again an “ $|\mathbf{E}|^2$ -profile” loss, associated with the predicted MMI perturbation pattern. Such tandem network architecture has recently proven to be very successful in photonics inverse design^{41,52–55}.

Our generator network is implemented as an encoder-decoder adaptation of a β -variational autoencoder (β -VAE)^{56–58}. This is done by adding a Kullback-Leibler-divergence term as a third loss to the training of the full model (“KL-loss” in figure 1b). This is a measure of how close the distribution of latent vectors is to a normal distribution. The latent vectors are the learned, compressed representations of the inverse target, indicated schematically in the center layer of the generator. The KL-loss forces the network to learn a smooth and continuous latent representation, which prevents that a small change in the design target unexpectedly leads to a radically different MMI pattern. It hence helps stabilizing the inverse design robustness.

The total loss to be minimized for the training of the generator is a weighted sum of the above described three losses. We found a weighting of 1 : 50 : 0.2 (pattern : $|\mathbf{E}|^2$: KL) to result in efficient training and good network performance. The KL-loss weight (0.2) corresponds to the β parameter in a β -VAE. In order to assure stable training of the variational generator, we found it necessary to apply a KL annealing scheme in which we slowly increase the KL weight from zero to its final value over the first couple of epochs⁵⁹. Furthermore we employ a mix of learning rate decay and batchsize growth to ensure full convergence of the training⁶⁰. More details as well as plots of training convergence are given in the supporting information figure S2.

Since the training of the ANNs relies on calculating gradients of the network output with respect to its parameters, the generated MMI patterns are necessarily gray-scale and need to be converted to a binary design. We do this by applying a threshold below which all gray-scale values are set to zero and the remaining positions are set to one. In order to avoid choosing an arbitrary threshold value, we use the pre-trained forward network to evaluate the resulting MMIs for a series of threshold values between 0 and 1 (in steps of 0.01). Eventually we choose the solution closest to the design target, using a mean square error metric. For more details and an analysis of the impact of varying the threshold, see supporting information section VIC and SI figure S3. Thanks to the very short prediction time of less than 0.5 ms per MMI (on a NVIDIA Quadro P6000 GPU), this does not create a considerable computational bottleneck. We still obtain the final inverse design in a time of the order of 50 milliseconds. We note that inference is also relatively efficient on CPU, the full process taking around 130 ms on a 3rd generation AMD Ryzen processor.

D. Iterative training

From initial investigations of inverse design performance (see supporting information figure S4 and “iter 0” in figure 2g which is discussed further below), we hypothesize that the homogeneity of the training data, which comprises MMIs that are produced by the same data-generation algorithm, hinders the forward network’s ability to correctly generalize. Since the forward ANN plays a crucial role in the training of the generator ANN itself, the accuracy of the forward ANN poses a fundamental problem for the inverse design. We therefore decided to implement an iterative training scheme based on the extension of the training dataset with ANN-generated patterns. This helps to improve the diversity of the training data and incorporates a self-correcting feedback effect into our network training scheme, thereby teaching the network in a “self-consistent” way to avoid its previous errors. Similar training techniques have been recently proposed for related inverse design problems^{62,63}.

In our iterative scheme depicted in Figure 1a, after each training cycle (steps 1-3), we use the generator network in step 4 to produce a dataset of 10000 random, inverse designed perturbation patterns. For details on the randomized inverse target generation, see the supporting information section VIA. We simulate the new inverse designed patterns by FDTD and combine these results with the previous training set, effectively closing the loop between step 4 and step 1. On the combined data, we train again first the forward ANN and then the generator network in the same way as described above. Through the ANN-generated, FDTD-simulated data, any features that were previously poorly modeled are fed back into the ANN model, thereby improving its performance. For all models in this work, we repeated the iterative data generation process over 6 full cycles. Detailed statistics of the networks during this iterative process can be found in the supporting information figures S4 and S5 for the 1×2 and 3×3 MMI cases, respectively.

E. Benchmark against experimental 1×2 MMI

In order to benchmark the simulations and neural network performance against an actual device, we consider a simple 1-input by 2-outputs (1×2) MMI with etched perturbations of index contrast $\Delta n_{\text{eff}} = -0.71$. We use etched perturbations for this first test in order to benchmark the simulations and neural network performance against an actual device.

In figure 2, we show a comparison of the 1×2 MMI forward network with FDTD simulations and experimental data. Fig. 2a shows a scanning electron microscopy (SEM) image of the patterned MMI device. The device transmission, normalized to a reference straight waveguide, was measured to be -22.2 dB and -1.0 dB (at $\lambda_0 = 1550$ nm) for the top and bottom outputs respectively.

Figure 2b shows a perturbation-based raster-scan transmittance measurement,⁶¹ which has been demonstrated to yield a mapping that is proportional to the local electric field intensity integrated over the surface area of the perturbation when

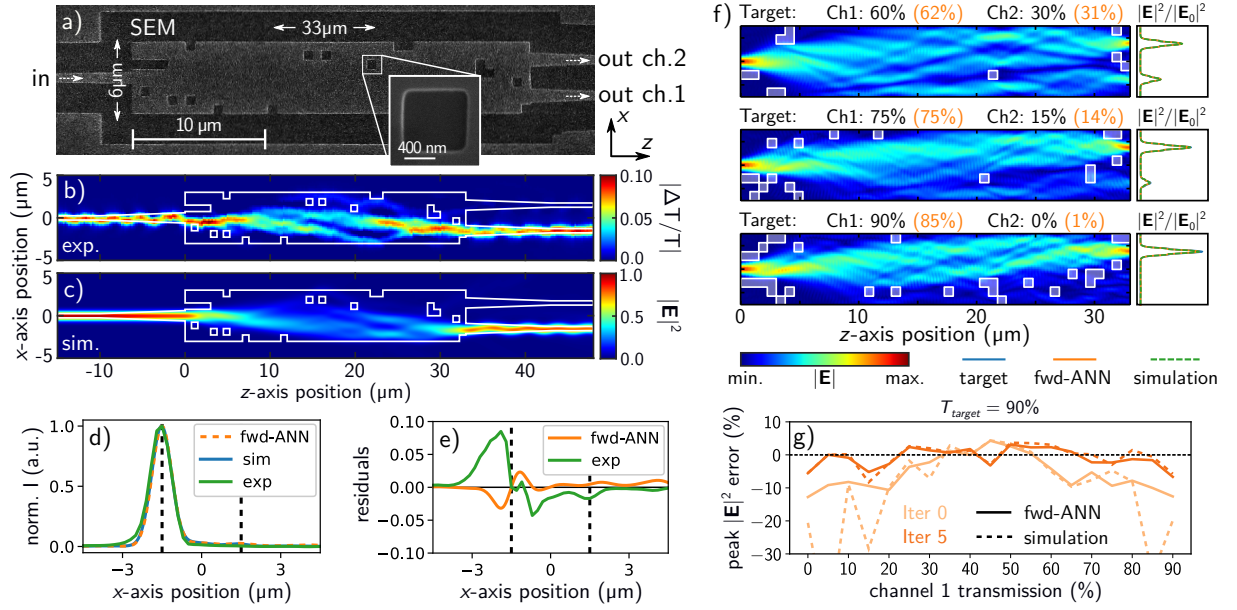


Figure 2. (a) MMI geometry, scale bars represent 10 μm (main) and 400 nm (zoom). (b) perturbation-based transmittance measurement.^{47,61} (c) field intensity distribution FDTD simulation. (d) illustration of the forward ANN which we test against the simulations and the experiment. (e) intensity-profile at output, obtained experimentally by perturbation-measurement (green), by FDTD simulation (blue) and with the forward neural network (orange). For the comparison, simulated and forward-ANN predicted intensity profiles in (c) and (d) are convoluted with a Gaussian of width 740 nm corresponding to the laser spot-size. (f) examples of inverse designed etched 1×2 MMI patterns (white overlay) with target 90% total transmittance and different splitting ratios. The electric field distributions of the etched patterns are calculated by FDTD simulations. Target and simulated transmittance values are given by the black and orange numbers, respectively. Right side of (f) shows the intensity profiles along the output waveguides for design target (solid blue lines), forward-ANN prediction (solid orange lines) and FDTD simulation (dashed green lines). (g) Inverse design MMI peak intensity error for forward-ANN prediction (solid line) and simulation (dashed line) versus target with respect to different channel splitting ratios and a total device transmittance of 90%. Pale and dark colors correspond to initial and final network iterations, respectively.

the transmittance is nearly unity⁴⁷. This can be seen in the qualitative comparison to the electric field simulation shown in Fig. 2c.

In figure 2d-e we compare the normalized intensity profiles across the output waveguides obtained by experiment (green lines) and FDTD simulation (blue lines) and the forward ANN (orange lines), showing an excellent agreement. Overall this result demonstrates the feasibility of an inverse-design ANN to mold the flow of light in integrated photonic devices via optical perturbations, in agreement with earlier studies¹⁹.

Figure 2f shows selected examples of inverse designed patterns from the final iteration of the generator, showing an excellent performance of the ANN-designed MMI patterns in comparison to FDTD simulated intensity-profiles. Target and simulated transmittance values are indicated by black and orange labels in the panels. The average error between design-target and FDTD simulation is generally as low as a few percent. Additional examples of the symmetrical cases are shown in the supporting information figure S6.

Finally, figure 2g compares the peak intensity error between the initial and final iterations of the ANNs for all splitting ratio cases in steps of 5% and a total target transmittance of 90%. Note that not only is the performance of inverse designed patterns significantly better after 5 iterations, but also that the agreement between the forward-ANN prediction and

simulated response is improved.

III. DESIGN OF TRANSMISSION MATRIX IN A MULTI-PORT MMI

A. Intensity-only transmission matrix design

The 1×2 MMI discussed so far reflects the simplest possible device configuration and corresponds to the case of a variable power splitter. Having successfully achieved the optimization of this case, we are now in a position to move to the more complex case of multi-port transmission matrices. In the following we test our inverse ANN on the example of a 3×3 MMI. Instead of using permanent etched perturbations, we consider in the following weaker index-contrast perturbations, $\Delta n_{\text{eff}} = -0.25$, corresponding to the optically induced perturbations of a recently demonstrated experimental configuration¹⁹. The effects of individual smaller perturbations is reduced and therefore more perturbations can be added to shape the light flow without strongly affecting the device throughput. Furthermore, we choose the 3×3 MMI to have the same 33 μm by 6 μm footprint as we use for the 1×2 devices. These dimensions result in very poor unperturbed splitting performance, rendering the design-problem

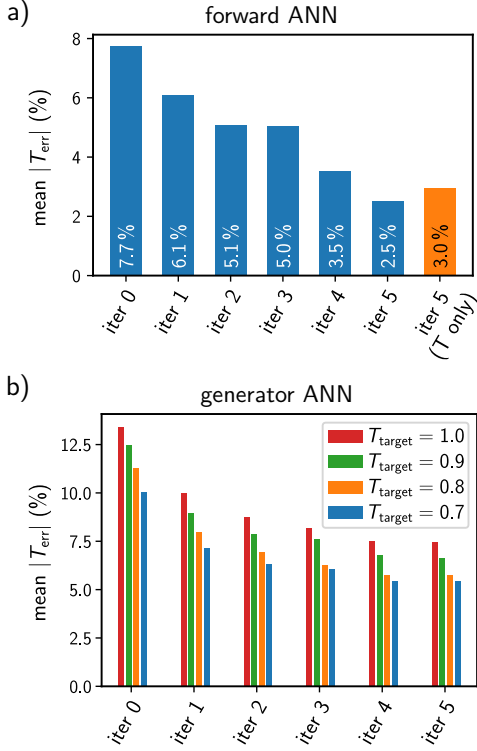


Figure 3. (a,b) 3×3 MMI mean absolute transmittance error between the ANN and simulations for the five consecutive data-generation iterations, calculated on a fixed test dataset. (a) forward network error: ANN-prediction vs. simulation. Iteration 5 (the full training set) is also compared to an ANN of identical architecture, where the last layer is replaced by a fully connected layer returning only the total transmittance of each port instead of the intensity profile. (b) inverse network error: design-target vs. simulation of suggested inverse-design. Total transmittance of the design targets is fixed to 70% (blue), 80% (orange), 90% (green) and 100% (red).

even more challenging, but maintain the compact device footprint. We note that a conventional 3×3 symmetric MMI splitter, i.e. with equal splitting ratios for all port couplings, and the same $6 \mu\text{m}$ width requires an $88 \mu\text{m}$ device length⁶⁴. Such an increased device length renders the transmission more sensitive to small fabrication imperfections and in fact may not be required when using the weak perturbation approach to shape the transmission matrix, provided that the device length is large enough to accommodate the required perturbation patterns.

For this geometry, we first test the performance of both the forward and the inverse networks after each data generation and training cycle. Each such cycle is done on a set of 1000 cases, generated in the same way as during step 4 of the iterative training scheme. For the inverse ANN, we furthermore split our test targets into four different total transmittance ranges with values of 70%, 80%, 90% and 100%, therefore 250 target transmission matrices are considered for each throughput value. The generated patterns are subsequently simulated by FDTD and results are compared to the target matrices. The accuracy of the forward and inverse ANNs

compared to FDTD simulations is shown in figure 3a and b as function of the training iteration number. There is a clear improvement in performance with subsequent iterations. After the iterative training, both the forward and inverse ANNs have experienced significant performance improvements of a factor $3\times$ and $2\times$, respectively, compared to the initial dataset of iteration 0.

In figure 3a we additionally compare the intensity-profile predicting forward-network with an ANN that predicts only the “bare” T values in each port (orange bar on the very right). We find that training a network on the full spatial intensity distribution instead of the per-port transmittance values leads to a slightly more accurate model (2.5% vs. 3%). While this is not a dramatic difference, the inverse training crucially relies on the forward ANN loss, and hence obtaining the best possible accuracy for the forward network is essential for the performance of the inverse network. Therefore, we use the full-profile in this work, but the technique should work also with ANNs operating on transmittance values only.

In figure 4, we show examples of transmission matrix design. The examples under study are part of an important class of operations, namely those that permute the device outputs, routing light from each input channel to a separate output port. The routing can then be dynamically controlled via the application of appropriate patterns. Figure 4a shows how the transmission matrix is constructed from the transmission for each independent input for a particular device. We consider here the intensity transmission matrix between the input and forward-output channels, which is depicted by a color-array, in which each column corresponds to an input channel and each row to an output channel. The possibility of obtaining a full, complex transmission matrix is discussed further below. We do not consider reflections back into the input ports in the matrix, which are generally weak for the regime under study¹⁹ and which are therefore considered as part of the overall insertion loss of the device. Figure 4b compares the target and generated matrices T_{tar} and T_{sim} , where the error matrix T_{err} gives the differences. We see an overall reduction in the total intensity of the design versus target total transmission, which was set to 80% (approx. -1 dB) in this case. Apart from this overall insertion loss, the generator ANN successfully manages to define devices that reproduce the target transmission matrix, as can be seen by the FDTD simulations in figure 4c.

In the supporting information figure S7, we show further examples of inverse design for many different input/output routing configurations, demonstrating a very robust and flexible performance of the inverse design ANN. We note, that the statistics in figure 3b are based on randomly generated, arbitrary transmission states, further demonstrating the flexibility of the method.

B. Latent space operations

During training, the generator network learns a compressed representation of the design targets – the so-called latent representation. Using the above discussed “KL-loss” forces the network to ensure that the latent vectors follow a normal

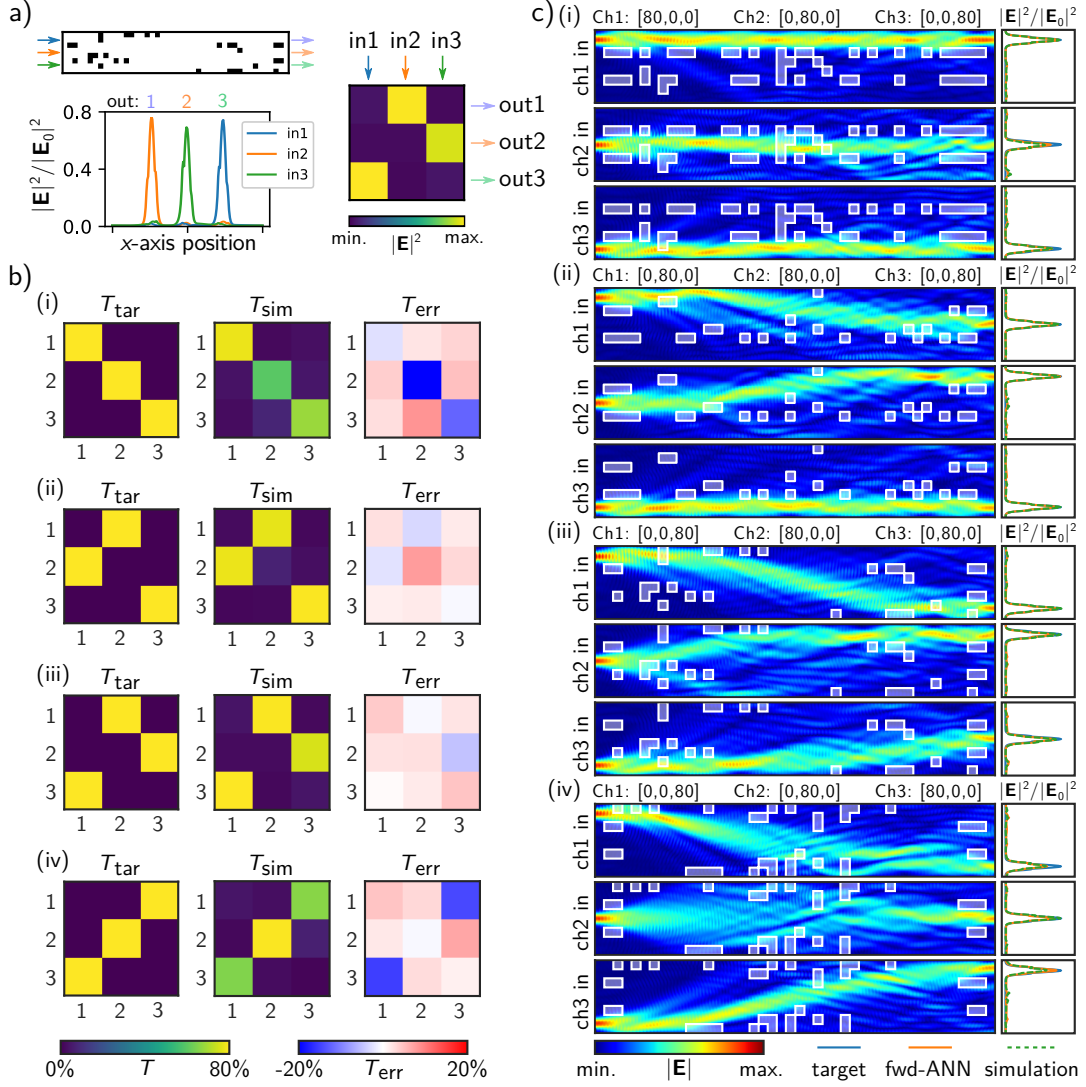


Figure 4. (a) illustration of the used color-array representation. The total transmitted intensity for each input/output port couple is represented by a color-code in a small 3×3 color-array. Thus, this array represents the intensity transmission matrix of the device. (b,c) examples (i-iv) of designed 3×3 transmission matrices (b) and corresponding simulated electric field distributions of the patterned MMI calculated using FDTD (c), for fundamental mode input in the three respective channels. Examples have a target of 80% total transmittance for each input channel. Intensity transmission matrices are shown for target (T_{tar}), simulated design (T_{sim}) and error (T_{err}). Right side of (c) shows the intensity profiles along the output waveguides for design target (solid blue lines), forward-ANN (solid orange lines) and FDTD simulation (dashed green lines).

distribution⁵⁶. In this case, it is possible to perform arithmetic operations such as interpolation, addition or subtraction in the latent space. Through the interpolation between design targets we can, for instance, obtain MMIs with a smooth transition in their physical behavior, and hence between their transmission matrices. We illustrate this for several examples in figure 5.

In figure 5a we interpolate the latent vectors between pairs of generated patterns corresponding to different intensity transmission matrices. The simulated transmission matrices show that the ANN indeed learned a smooth latent representation. Further examples can be found online as movies, in which the transition between two physical solutions is animated along the interpolation ([Link to Nic's visualizations](#)). In figure 5b we show three examples of latent space summa-

tion and subtraction. The latent vectors of two designs are added or subtracted and the resulting latent vector is fed into the generator. This yields an MMI with accordingly combined or subtracted transmission matrix. As can be seen when comparing the MMI patterns of the latent-operations, the solutions do not correspond to a simple summation of, or an interpolation between the MMI patterns. We note that in case of the latent operations we cannot use the forward network in order to find the best threshold value to get a binary representation of the generated pattern, in the examples of figure 5 we therefore used the threshold of the respective “design 1”.

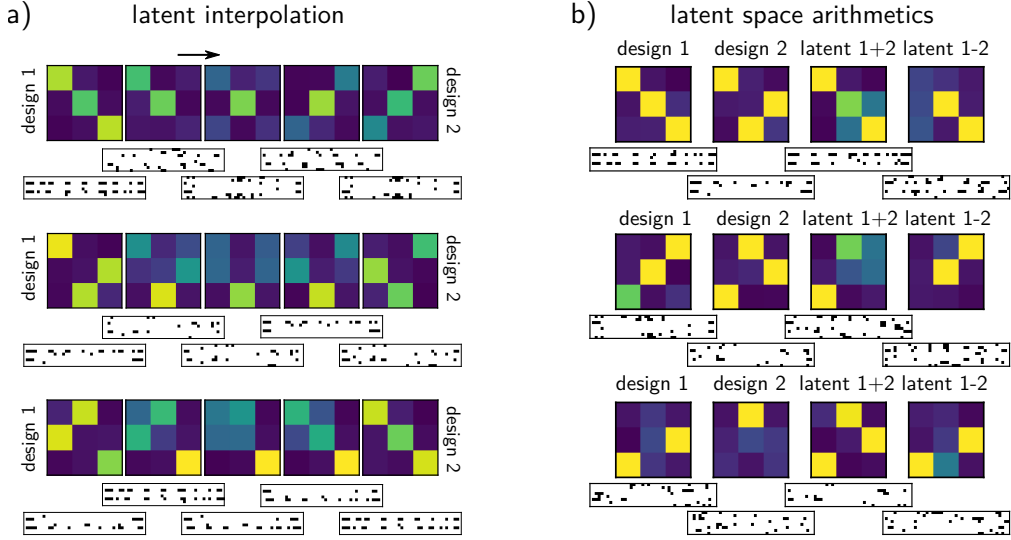


Figure 5. Variational generator latent space operations. (a) solutions with a smooth transition of the physical response obtained by linear interpolation of the latent space vector between two inverse-designed MMI patterns (“design 1” at the left and “design 2” at the right end). (b) examples of simple arithmetic operations in the latent space of our variational generator ANN. This is demonstrated here for three examples of latent-space summation and subtraction. All shown color-scale transmission matrices are calculated by FDTD simulations, color coding is the same as in figure 4.

C. Phase-aware inverse design

So far, the ANNs in our study were operating only with the transmitted intensity. This is sufficient when considering for example routers where intensity is routed between ports with little cross-interference, or when considering mutually incoherent input channels where coherent cross-interference at the outputs can be neglected. For many applications, such as coherent datacommunication, quantum optics, and analog computing photonic devices^{8,18}, full control over the complex transmission matrix is required, and hence over both amplitude and phase of the electric field at the outputs. An overall phase offset may be introduced by adding additional phase shifters behind the output ports but this approach is insufficient for the most general case where each output port is coupled to each input with a completely independent phase. Therefore it is of great interest to include phase control into the transmission matrix design itself. To introduce full phase control, we start by training the forward ANN on the prediction of the complex electric fields across the output waveguides obtained using FDTD. The prediction accuracy is found to be similar to the intensity-only forward ANN. However instead of the M ANN-outputs in the intensity-only case, where M is the number of MMI input channels, the complex fields predicting forward ANN returns $4 \times M$ profiles, where the factor 4 reflects the real and imaginary parts of E_x and E_z . Note that in our 2D simulations using TE polarized light, the out-of-plane component $E_y = 0$.

For the phase-aware generator model, we assume that $E_x \gg E_z$, in which case the phase of the transverse mode can be approximated by $\vartheta = \arctan(\text{Re}(E_x)/\text{Im}(E_x))$. To avoid having the spatial profiles of complex field amplitudes as design targets, we use for the generator ANN only the real and imagi-

nary parts of the field at the location of peak intensity in the respective output waveguide, $\text{Re}(E_{x,\text{peak } l})$ and $\text{Im}(E_{x,\text{peak } l})$. The ANNs are trained on the design dataset from iteration 5 obtained from the intensity-only 3×3 ANN.

As a proof-of-principle of design of the complex transmission matrix, we select a fixed intensity matrix corresponding to the pattern shown in the left panels of Fig. 6a. Next we define the real and imaginary part of the amplitude such that the output phase is rotated through a 2π cycle with respect to the input. The complex amplitude for the MMI transmission is illustrated in Fig. 6a with a hue / brightness based color-coding (see legend on the right). In this example, our target assumes a constant phase for all output channels. In the SI we also show the inverse design of phase-rotation for more complex challenges, where we furthermore add a relative phase of $\pi/6$ between each non-zero output, in order to show independent control of the phase of each port. This and more examples are given in the supporting information figure S8.

The MMI patterns obtained by the generator ANN are shown in Fig. 6b, the corresponding simulated complex amplitude transmission matrices are depicted in Fig. 6a in the right panels. To improve the readability of the figure, we clip the color-coding to white for output-channels with transmitted intensities $< 5\%$.

We observe that in general, the inverse network succeeds in finding MMI patterns which reproduce both the desired transmission pattern as well as the target phase at the outputs. Examining the examples labeled (i) to (iii) in Fig. 6c, we observe that the ANN uses two basic strategies to achieve phase design. The first strategy consists of the introduction of perturbations into or close to the light path, which induces a phase shift due to the effective index change of the propagation medium. The second strategy involves the modification

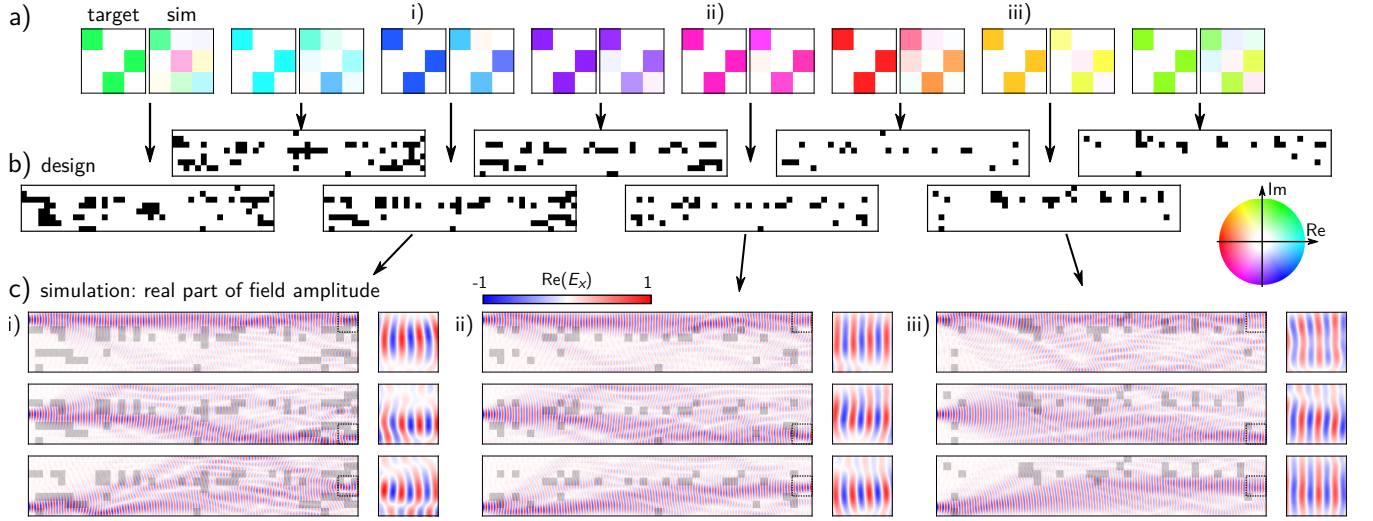


Figure 6. Example of design of a complex transmission matrix. An inverse network is trained using internally a full-field forward network and taking as inverse target the real and imaginary part of the complex field amplitude E_x . This allows to tailor not only the transmitted intensity, but also the relative phase of each transmission channel. (a) left: design target. We fix an intensity pattern and rotate the overall relative phase with respect to the input. Right: simulated complex transmission matrix for E_x . In order to increase the readability of the figure the colorcode is clipped to white for transmitted intensities $< 5\%$. (b) inverse designed perturbation patterns. (c) Real part of E_x along the full MMI for the examples i)-iii), where (ii) and (iii) are designed to have respectively a $\pi/2$ and π phase relative to (i). The colorscale is normalized to the input amplitude. From top to bottom: Input into upper, center and lower channel. The right subplots show a zoom of $2 \times 2 \mu\text{m}^2$ on the respective target output channel regions (indicated by a dashed square).

of the light path length. In the latter case, the ANN gains control over the phase by routing the light along shorter or longer paths, as can be observed in the center and bottom input cases in Fig. 6c.

In most cases, the phase-design ANN yields MMI patterns with reasonable qualitative performance. However we note that the agreement is quantitatively much worse than the above shown intensity-only examples. Controlling both phase and intensity is a far more complex problem than intensity only transmission design. Hence, full fine-control over the phase would probably require a larger MMI footprint and/or a larger number of stronger perturbations. While there are no principle technical constraints, larger MMIs and smaller perturbations would considerably increase FDTD simulation time and hence slow down data generation and the training phase considerably. Further studies are needed to explore this design space.

Despite the generally good qualitative performance, in certain cases the inverse-designed MMIs perform poorly where some channels suffer from low transmission. This can be seen for instance in the very left design target of figure 6, where the number of perturbations in the ANN design is very large. But also in these cases of poor inverse design performance, we observe that the forward network predictions are in very good agreement with the FDTD simulations. Therefore we conclude that the phase-design problems are not due to a poor generalization capability of the ANNs. Judging from the perturbation patterns (Fig. 6b and also several examples in supporting information Fig. S8), we believe that the problematic cases are at the very limit of what is possible within the relatively small MMI-size and with the limited amount and weak

index contrast of the perturbations. In case of the very left MMI in figure 6, the ANN adds a large amount of perturbations in an attempt to tune the phase to the desired angle. The high perturbation density however deteriorates severely the general performance of the device, suppressing the transmittance for input channels 2 and 3. In the MMI at the very right of figure 6, the ANN followed the opposite strategy and inserted as few perturbations as possible, obviously arriving at the other side of the limit of its working range. In agreement with those observations we find that the phase-design MMI patterns with the largest and with the least number of perturbations usually perform worst (c.f. supporting information Fig. S8). The deterioration of the MMI performance with too many or too few perturbations can also be seen in the supporting information figure S3.

IV. DISCUSSION

Our investigations have shown that neural networks can be trained to generate arbitrary transmission matrices by designing a corresponding perturbation pattern in a multi-mode device. One of the main challenges in this work was to give the network sufficient examples of high-throughput solutions, which are the desirable patterns for many applications but are also sparsely distributed in the overwhelmingly large design space.

A valid question is whether the neural network actually conceives a generalized representation or if it merely ‘memorizes’ results through an elaborate lookup table functionality. To determine whether the neural networks generalize well

and find original solutions beyond what is available in the simulated database, we compare the inverse designs of our test-cases with the training data. In table S1 presented in the supporting information we summarize the average number of non-identical perturbations between the ANN designs and the respective most similar patterns in the training data, both for the ANNs trained on only the initial data set as well as for the ANN after the iterative training. In case of the 1×2 MMIs we observe that after the iterative training, for any inverse designed pattern there can be found in most cases an almost identical device in the training set (< 2 non-matching perturbation positions). We attribute this overlap to the lower complexity of the 1×2 MMI problem, being essentially just a power-splitter.

For the more complex 3×3 MMIs on the other hand, we found that the ANN does not simply act as a lookup table and the generated designs are usually very different from the closest design in the dataset (average mismatch count 17.1 ± 12.2 after iterative training). In this case, the number of non-matching perturbations is not decreasing significantly during the iterative training. Clearly, in order to solve the rather complex 3×3 transmission matrix problem, it is necessary for the ANN to generalize well, in order to be able to invent original solutions.

For the phase-aware transmission matrix design, we have shown the potential as well as the limitations for the current geometry. Overall the neural network is capable of including phase in the transmission matrix. In the supporting information figure S8 we show additional examples for the phase-design network, including an example with the same intensity-target as in figure 6, but with independent phases for the different output channels as compared to the same overall phase. A similar performance is achieved, indicating that there is no constraint in the design of the output phases for each port.

Again, as in the above discussed examples, the cases of poor MMI performance can usually be explained by insufficient design margins in our chosen geometry which limits the ANN in tuning the phase over a full 2π range. Larger device geometries and/or stronger perturbations are expected to provide access to a larger design space, however, this needs to be traded off against transmission losses which increase with the number of perturbations and their strength. While current work has been aiming to achieve high-throughput solutions, for real-world applications the extinction ratio and crosstalk between ports will be additional important parameters, which cannot be easily implemented as optimization targets in the neural network approach. We anticipate that these additional performance figures can be fine tuned, for example by further improving the solutions generated by the network using multi-objective iterative forward optimization.

At this stage, each new design geometry, i.e. variations in

length and width of the MMI and perturbation strength, requires a completely new dataset to be generated which limits the range of parameters that can be easily explored. Future work will look at expanding the geometrical design space, including increasing the number of input and output channels to ultimately achieve large scale multi-input, multi-output (MIMO) programmable devices. Such explorations may be able to achieve further improvements in the neural network approach, including strategies such as transfer learning to improve the agility of the neural approach in adapting to variations in device geometries.

V. CONCLUSION

In conclusion, we have developed a method of designing universal optical components based on weak scattering perturbations in a multimode waveguide. Arbitrary transmission matrices were generated using a deep learning neural network. By directing the neural network toward high throughput solutions using an iterative data generation scheme, we have achieved reliable results with a mean-square error $< 5\%$ compared to numerical simulations. The neural network approach, while computationally demanding to train, is fast (< 50 ms) in the generation of solutions and hence could be used in real-time applications. We have furthermore shown that independent phase control can be readily implemented, where we have seen some of the limitations in the limited design space under study. This work opens up a conceptually new approach for programmable photonic systems based on weak and forward multiple scattering effects. Such integrated devices with a freely tunable transmission matrix, at length scales far below that of interferometer meshes, will find application as ultra-compact universal optical elements for applications in optical routing, programmable weight banks, or analogue computing.

ACKNOWLEDGMENTS

We thank the NVIDIA Corporation for the donation of a Quadro P6000 GPU used for this research. Silicon photonic waveguides were manufactured through the UK Cornerstone open access Silicon Photonics rapid prototyping foundry through the EPSRC grant EP/L021129/1. P.R.W. acknowledges support by the German Research Foundation (DFG) through a research fellowship (WI 5261/1-1). OM acknowledges support through EPSRC grant EP/M009122/1. All data supporting this study are openly available from the University of Southampton repository (DOI: 10.5258/SOTON/XXXXXXXXXX).

* e-mail : pwiecha@laas.fr

† e-mail : o.muskens@soton.ac.uk

¹ W. Bogaerts, A. Rahim, Programmable photonics: An opportunity for an accessible large-volume pic ecosystem. *IEEE Journal*

of Selected Topics in Quantum Electronics **26**, 1-17 (2020).

² X. Chen, M. M. Milosevic, D. J. Thomson, A. Z. Khokhar, Y. Franz, A. F. J. Runge, S. Mailis, A. C. Peacock, G. T. Reed, Post-fabrication phase trimming of mach–zehnder inter-

- ferometers by laser annealing of germanium implanted waveguides. *Photon. Res.* **5**, 578–582 (2017).
- ³ L. Zhuang, C. G. H. Roeloffzen, M. Hoekman, K.-J. Boller, A. J. Lowery, Programmable photonic signal processor chip for radiofrequency applications. *Optica* **2**, 854–859 (2015).
 - ⁴ R. Stabile, A. Albore-Mejia, A. Rohit, K. A. Williams, Integrated optical switch matrices for packet data networks (2016).
 - ⁵ Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, M. Soljacic, Deep learning with coherent nanophotonic circuits. *Nature Photonics* **11**, 441–446 (2017).
 - ⁶ N. C. Harris, G. R. Steinbrecher, M. Prabhu, Y. Lahini, J. Mower, D. Bunandar, C. Chen, F. N. Wong, T. Baehr-Jones, M. Hochberg, S. Lloyd, D. Englund, Quantum transport simulations in a programmable nanophotonic processor. *Nature Photonics* **11**, 447–452 (2017).
 - ⁷ N. C. Harris, J. Carolan, D. Bunandar, M. Prabhu, M. Hochberg, T. Baehr-Jones, M. L. Fanto, A. M. Smith, C. C. Tison, P. M. Alsing, D. Englund, Linear programmable nanophotonic processors. *Optica* **5**, 1623 (2018).
 - ⁸ J. Wang, F. Sciarrino, A. Laing, M. G. Thompson, Integrated photonic quantum technologies (2020).
 - ⁹ D. A. B. Miller, Perfect optics with imperfect components. *Optica* **2**, 747–750 (2015).
 - ¹⁰ W. R. Clements, P. C. Humphreys, B. J. Metcalf, W. S. Kolthammer, I. A. Walmsley, Optimal design for universal multiport interferometers. *Optica* **3**, 1460–1465 (2016).
 - ¹¹ S. Rotter, S. Gigan, Light fields in complex media: mesoscopic scattering meets wave control. *Rev. Mod. Phys.* **89**, 015005 (2017).
 - ¹² A. P. Mosk, A. Lagendijk, G. Leroose, M. Fink, Controlling waves in space and time for imaging and focusing in complex media. *Nature Photonics* **6**, 283–292 (2012).
 - ¹³ V. Liu, Y. Jiao, D. A. B. Miller, S. Fan, Design methodology for compact photonic-crystal-based wavelength division multiplexers. *Opt. Lett.* **36**, 591–593 (2011).
 - ¹⁴ J. Jensen, O. Sigmund, Topology optimization for nanophotonics. *Laser & Photonics Reviews* **5**, 308–321 (2011).
 - ¹⁵ A. Y. Piggott, J. Lu, K. G. Lagoudakis, J. Petykiewicz, T. M. Babinec, J. Vučković, Inverse design and demonstration of a compact and broadband on-chip wavelength demultiplexer. *Nature Photonics* **9**, 374–377 (2015).
 - ¹⁶ L. F. Frellsen, Y. Ding, O. Sigmund, L. H. Frandsen, Topology optimized mode multiplexing in silicon-on-insulator photonic wire waveguides. *Optics Express* **24**, 16866–16873 (2016).
 - ¹⁷ T. W. Hughes, I. A. Williamson, M. Minkov, S. Fan, Wave physics as an analog recurrent neural network. *Science Advances* **5**, eaay6946 (2019).
 - ¹⁸ N. M. Estakhri, B. Edwards, N. Engheta, Inverse-designed metastructures that solve equations. *Science* **363**, 1333–1338 (2019).
 - ¹⁹ R. Bruck, K. Vynck, P. Lalanne, B. Mills, D. J. Thomson, G. Z. Mashanovich, G. T. Reed, O. L. Muskens, All-optical spatial light modulator for reconfigurable silicon photonic circuits. *Optica* **3**, 396 (2016).
 - ²⁰ A. Brandstötter, A. Girschik, P. Ambichl, S. Rotter, Shaping the branched flow of light through disordered media. *Proceedings of the National Academy of Sciences* **116**, 13260–13265 (2019).
 - ²¹ I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning* (MIT Press, 2016).
 - ²² Y. LeCun, Y. Bengio, G. Hinton, Deep learning. *Nature* **521**, 436–444 (2015).
 - ²³ Y. Rivenson, Y. Zhang, H. Günaydin, D. Teng, A. Ozcan, Phase recovery and holographic image reconstruction using deep learning in neural networks. *Light: Science & Applications* **7**, 17141–17141 (2018).
 - ²⁴ Y. Jo, S. Park, J. Jung, J. Yoon, H. Joo, M.-h. Kim, S.-J. Kang, M. C. Choi, S. Y. Lee, Y. Park, Holographic deep learning for rapid optical screening of anthrax spores. *Science Advances* **3**, e1700606 (2017).
 - ²⁵ P. R. Wiecha, A. Lecestre, N. Mallet, G. Larrieu, Pushing the limits of optical information storage using deep learning. *Nature Nanotechnology* **14**, 237–244 (2019).
 - ²⁶ B. Han, Y. Lin, Y. Yang, N. Mao, W. Li, H. Wang, V. Fatemi, L. Zhou, J. I.-J. Wang, Q. Ma, Y. Cao, D. Rodan-Legrain, Y.-Q. Bie, E. Navarro-Moratalla, D. Klein, D. MacNeill, S. Wu, W. S. Leong, H. Kitadai, X. Ling, P. Jarillo-Herrero, T. Palacios, J. Yin, J. Kong, Deep Learning Enabled Fast Optical Characterization of Two-Dimensional Materials. *arXiv:1906.11220 [cond-mat, physics:physics]* (2019).
 - ²⁷ B. Rahmani, D. Loterie, G. Konstantinou, D. Psaltis, C. Moser, Multimode optical fiber transmission with a deep learning network. *Light: Science & Applications* **7**, 69 (2018).
 - ²⁸ N. Borhani, E. Kakkava, C. Moser, D. Psaltis, Learning to see through multimode fibers. *Optica* **5**, 960–966 (2018).
 - ²⁹ U. Kürüm, P. R. Wiecha, R. French, O. L. Muskens, Deep learning enabled real time speckle recognition and hyperspectral imaging using a multimode fiber array. *Optics Express* **27**, 20965–20979 (2019).
 - ³⁰ B. Shen, P. Wang, R. Polson, R. Menon, An integrated-nanophotonics polarization beamsplitter with $2.4 \times 2.4 \text{ Mm}^2$ footprint. *Nature Photonics* **9**, 378–382 (2015).
 - ³¹ B. Shen, R. Polson, R. Menon, Metamaterial-waveguide bends with effective bend radius $\ll \lambda$. *Optics Letters* **40**, 5750–5753 (2015).
 - ³² B. Shen, P. Wang, R. Polson, R. Menon, Integrated metamaterials for efficient and compact free-space-to-waveguide coupling. *Optics Express* **22**, 27175–27182 (2014).
 - ³³ J. S. Jensen, O. Sigmund, Topology optimization for nanophotonics. *Laser & Photonics Reviews* **5**, 308–321 (2011).
 - ³⁴ T. Feichtner, O. Selig, M. Kiunke, B. Hecht, Evolutionary Optimization of Optical Antennas. *Physical Review Letters* **109**, 127701 (2012).
 - ³⁵ P. R. Wiecha, A. Arbouet, C. Girard, A. Lecestre, G. Larrieu, V. Paillard, Evolutionary multi-objective optimization of colour pixels based on dielectric nanoantennas. *Nature Nanotechnology* **12**, 163–169 (2017).
 - ³⁶ K. Xu, L. Liu, X. Wen, W. Sun, N. Zhang, N. Yi, S. Sun, S. Xiao, Q. Song, Integrated photonic power divider with arbitrary power ratios. *Optics letters* **42**, 855–858 (2017).
 - ³⁷ H. Jia, T. Zhou, X. Fu, J. Ding, L. Yang, Inverse-design and demonstration of ultracompact silicon meta-structure mode exchange device. *Acs Photonics* **5**, 1833–1838 (2018).
 - ³⁸ M. van Niekerk, J. A. Steidle, G. A. Howland, M. L. Fanto, N. Soures, F. T. Zohora, D. Kudithipudi, S. F. Preble, *Quantum Information Science, Sensing, and Computation XI* (International Society for Optics and Photonics, 2019), vol. 10984, p. 109840J.
 - ³⁹ Y. Xie, T. Huang, Q. Ji, M. Yang, J. Wang, X. Tu, Z. Cheng, G. Xu, Q. Wei, Y. Wu, P. P. Shum, Design of an arbitrary ratio optical power splitter based on a discrete differential multiobjective evolutionary algorithm. *Applied Optics* **59**, 1780–1785 (2020).
 - ⁴⁰ Z. Liu, D. Zhu, S. P. Rodrigues, K.-T. Lee, W. Cai, Generative Model for the Inverse Design of Metasurfaces. *Nano Letters* **18**, 6570–6576 (2018).
 - ⁴¹ Z. Liu, L. Raju, D. Zhu, W. Cai, A Hybrid Strategy for the Discovery and Design of Photonic Nanostructures. *arXiv:1902.02293 [physics]* (2019).
 - ⁴² C. C. Nadell, B. Huang, J. M. Malof, W. J. Padilla, Deep learning for accelerated all-dielectric metasurface design. *Optics Express*

- 27, 27523–27535 (2019).
- 43 M. H. Tahersima, K. Kojima, T. Koike-Akino, D. Jha, B. Wang, C. Lin, K. Parsons, Deep Neural Network Inverse Design of Integrated Photonic Power Splitters. *Scientific Reports* **9**, 1368 (2019).
 - 44 Y. Tang, Y. Tang, K. Kojima, T. Koike-Akino, Y. Wang, P. Wu, M. Tahersima, D. Jha, K. Parsons, M. Qi, *Optical Fiber Communication Conference (OFC) 2020 (2020), Paper Th1A.1* (Optical Society of America, 2020), p. Th1A.1.
 - 45 D. Thomson, Y. Hu, G. Reed, J.-M. Fedeli, Low loss mmi couplers for high performance mzi modulators. *IEEE Photonics Technology Letters* **22**, 1485–1487 (2010).
 - 46 E. Silberstein, P. Lalanne, J.-P. Hugonin, Q. Cao, Use of grating theories in integrated optics. *JOSA A* **18**, 2865–2875 (2001).
 - 47 K. Vynck, N. J. Dinsdale, B. Chen, R. Bruck, A. Z. Khokhar, S. A. Reynolds, L. Crudgington, D. J. Thomson, G. T. Reed, P. Lalanne, O. L. Muskens, Ultrafast perturbation maps as a quantitative tool for testing of multi-port photonic devices. *Nature Communications* **9**, 2246 (2018).
 - 48 P. Lalanne, M. Besbes, J. Hugonin, S. Van Haver, O. Janssen, A. Nugrowati, M. Xu, S. Pereira, H. Urbach, A. Van de Nes, *et al.*, Numerical analysis of a slit-groove diffraction problem. *Journal of the European Optical Society-Rapid Publications* **2** (2007).
 - 49 L. Chrostowski, M. Hochberg, *Silicon photonics design: from devices to systems* (Cambridge University Press, 2015).
 - 50 C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv:1602.07261 [cs]* (2016).
 - 51 D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]* (2014).
 - 52 D. Liu, Y. Tan, E. Khoram, Z. Yu, Training Deep Neural Networks for the Inverse Design of Nanophotonic Structures. *ACS Photonics* **5**, 1365–1369 (2018).
 - 53 L. Xu, M. Rahmani, Y. Ma, D. A. Smirnova, K. Z. Kamali, F. Deng, Y. K. Chiang, L. Huang, H. Zhang, S. Gould, D. N. Neshev, A. E. Miroshnichenko, Enhanced Light-Matter Interactions in Dielectric Nanostructures via Machine Learning Approach. *arXiv:1912.10212 [physics]* (2019).
 - 54 S. So, J. Mun, J. Rho, Simultaneous inverse design of materials and parameters of core-shell nanoparticle via deep-learning: Demonstration of dipole resonance engineering. *arXiv:1904.02848 [physics]* (2019).
 - 55 W. Ma, F. Cheng, Y. Xu, Q. Wen, Y. Liu, Probabilistic Representation and Inverse Design of Metamaterials Based on a Deep Generative Model with Semi-Supervised Learning Strategy. *Advanced Materials* **31**, 1901111 (2019).
 - 56 I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, β -VAE: Learning basic visual concepts with a constrained variational framework. *ICLR conference* (2017).
 - 57 C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, A. Lerchner, Understanding disentangling in β -VAE. *arXiv:1804.03599 [cs, stat]* (2018).
 - 58 D. P. Kingma, M. Welling, An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning* **12**, 307–392 (2019).
 - 59 S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, S. Bengio, Generating Sentences from a Continuous Space. *arXiv:1511.06349 [cs]* (2016).
 - 60 S. L. Smith, P.-J. Kindermans, C. Ying, Q. V. Le, Don’t Decay the Learning Rate, Increase the Batch Size. *arXiv:1711.00489 [cs, stat]* (2018).
 - 61 R. Bruck, B. Mills, B. Troia, D. J. Thomson, F. Y. Gardes, Y. Hu, G. Z. Mashanovich, V. M. N. Passaro, G. T. Reed, O. L. Muskens, Device-level characterization of the flow of light in integrated photonic circuits using ultrafast photomodulation spectroscopy. *Nature Photonics* **9**, 54–60 (2015).
 - 62 F. Wen, J. Jiang, J. A. Fan, Progressive-Growing of Generative Adversarial Networks for Metasurface Optimization. *arXiv:1911.13029 [physics]* (2019).
 - 63 A.-P. Blanchard-Dionne, O. J. F. Martin, Successive Training of a Generative Adversarial Network for the Design of an Optical Cloak. *arXiv:2005.08832 [physics]* (2020).
 - 64 L. B. Soldano, E. C. Pennings, Optical multi-mode interference devices based on self-imaging: principles and applications. *Journal of Lightwave Technology* **13**, 615–627 (1995).

VI. SUPPORTING INFORMATION

A. Data-generation

For the ANN’s ability to generalize well, we need to generate random transmission matrix targets for both the initial training dataset, iterative training sample generation, as well as for the statistical testing of the ANN performance. In this procedure, we first define a target device total transmittance, taking into account that our platform usually produces losses in the order of 10 – 30%.

For the optimization of patterns in initial training dataset, the total target transmittance is fixed to 100% and the splitting ratio between output ports is randomized. Figure S1a illustrates an example of pattern generation for the iterative optimization of coupling towards a single output of a 1×2 MMI using the a-FMM solver. The colored circles correspond to the perturbation patterns, and their FDTD simulated electric field distributions and intensity profiles shown in Figure S1b.

When using the ANN to generate new patterns, the total transmittance target has a direct effect on the networks performance. Therefore, each input channel is assigned a randomized total transmittance between 70–100%, which is then randomly distributed over the different output ports. Due to reciprocity the transmission matrices need furthermore to be unitary, which we obtain from the target matrix elements by also re-normalizing the row vectors (output-channels) in case their norm is > 1 .

B. ANN details and hyperparameters

We use a tandem network consisting of two networks, as shown in figure 1b in the main text. The first part is a forward network (green box), which is pre-trained in a first step to predict the MMI transmission. The second constituent is the variational generator network (blue box), which shall generate MMI patterns that fulfil the desired transmittance target. Both networks are encoder-decoder ResNet models^{S50}, consisting of down- and up-sampling residual blocks. Each residual block consists of three repetitions of consecutive convolutional, batch normalization and leaky ReLU layers. After each residual block, a max pooling operation is performed to divide the layer size by two. After each max pooling the number of filters is doubled, following the commonly used design

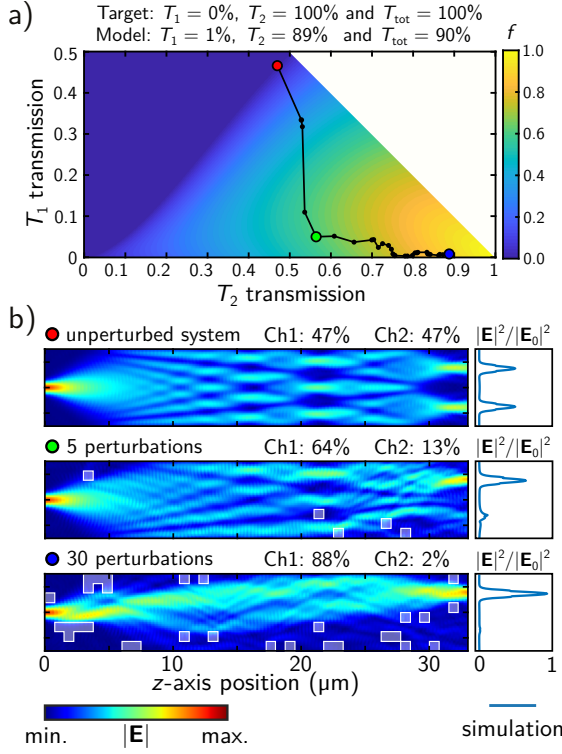
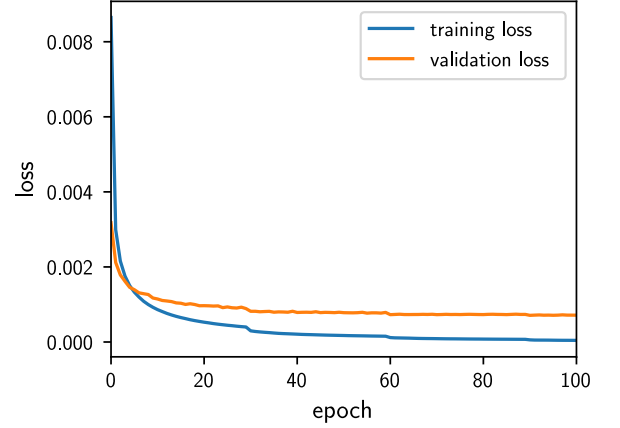


Figure S1. (a) optimisation of a perturbation pattern for a specific target transmission state, here Ch.1–0% and Ch.2–100%. (b) iteratively designed randomized MMI patterns with corresponding FDTD simulated $|E|^2$ field distributions (colormaps) and intensity profiles across the output waveguides (blue lines in right subplots). Coloured circles in (a) correspond to the accordingly labeled simulated patterns in (b).

principle. The output layer of both the forward network and of the generator uses a sigmoid activation to match the numerical range of the normalized intensities or field amplitudes, respectively the binary MMI patterns (zero or one). In the decoders, zero padding is used when necessary to reconstruct a network output of correct dimensions. This is indicated as “(+1)” in the labels indicating layer dimensions in figure 1b of the main text. The kernel sizes are in the 1D case for decreasing layer dimension: 5, 4, 3 and 3. In the 2D parts of the networks the kernel sizes are (4×4) , (3×3) , (3×3) and (2×2) . In the forward network, a dense layer of 256 neurons interconnects the 2D encoder and the 1D decoder. In the case of the variational generator we use a latent vector representation of length 64. We use the common reparametrization trick^{S58} in order to be able to backpropagate through the latent representation despite the random sampling of latent vectors z , which is required to calculate the KL-loss. To this end, the network is built such that it represents a latent vector using its mean μ and variance σ , which are differentiable for backpropagation. From those values we can then draw random samples assuming a normal distribution $N(\mu, \sigma)$.

We first train only the forward network using the ADAM solver^{S51} with a learning rate of 5×10^{-5} and a batchsize (BS) of 32. We increase the BS by a factor of 2 after 25 epochs

a) forward network



b) inverse network

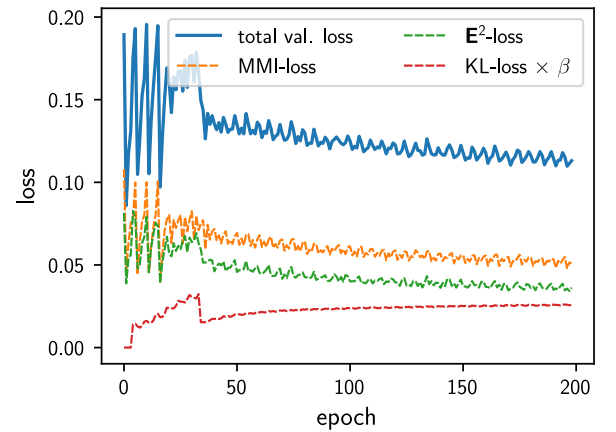


Figure S2. Example of loss convergence during (a) forward and (b) generator network training.

and by another factor of 2 after 60 epochs.^{S60} Once the forward network was trained for 100 epochs, we fix its parameters and train the full model, using a combination of three loss functions: first the MMI pattern loss (comparing generated MMI and training sample MMI patterns with a mean-square error (MSE) metric), second the $|E|^2$ -profile loss (comparing the pre-trained forward network’s prediction of the generated MMI with the training sample’s intensity profile on an MSE metric), and third the KL-loss to assure a uniform latent space^{S58}. The three losses are weighted by 1 : 50 : 0.2, which we determined to work well. Furthermore, in order to assure stable training, we found it to be necessary to ramp up the weight of the KL loss from zero to its final value 0.2 over the first 30 epochs (“KL-annealing”^{S59}). We use the same learning rate and BS increase scheme for the forward network alone as well as for the full model.

Typical convergence plots of loss vs. training epoch are shown in figure S2a and S2b for the forward, respectively generator networks.

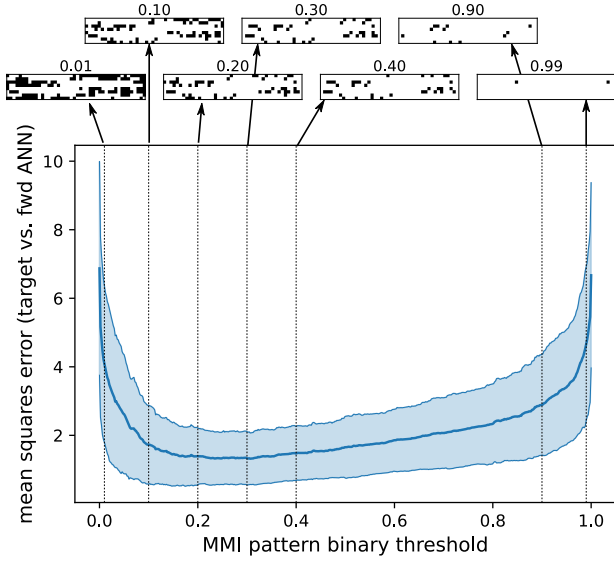


Figure S3. Mean squares error between design target and forward network predicted MMI intensity profile averaged for 150 inverse designed test MMIs as function of the threshold value. Thick blue line is the mean of the test-set, the shaded area corresponds to the standard deviation. A selected MMI is shown at the top, converted from grey-scale to binary using different threshold values (indicated with the label at the top).

C. Binary pattern threshold determination

Deep learning network training is an optimization process relying on the gradients of the ANN output with respect to the internal network parameters. Hence, the output of an ANN is necessarily continuous, whereas our MMI designs are binary. Therefore we need to apply a threshold value to transform the greyscale ANN designs into binary perturbation patterns. As described in the main text, the best threshold can be obtained using the forward network for testing of the MMI performance. Practically, various levels of perturbations could be introduced experimentally via different etch heights, however, this would require additional fabrication steps. With regards to the weaker optical perturbations considered, some intensity modulation could be applied to each projected pump spot to achieve a similar control over the refractive index change. In figure S3 we show the mean square error between target and ANN-designed MMI as function of the threshold value. The blue line is the average of 150 test cases, the shaded area corresponds to the standard deviation. At the top, perturbation patterns of a selected example are shown for several threshold values to illustrate the impact on the MMI. As can be seen, between threshold values around 0.2 and 0.4, the design error has a plateau-like minimum and the pattern is almost unchanged. If inverse-design speed should need to be further increased, the threshold could be fixed to a value in the center of this plateau, reducing the inverse design time from around 50 ms to the order of 1 ms.

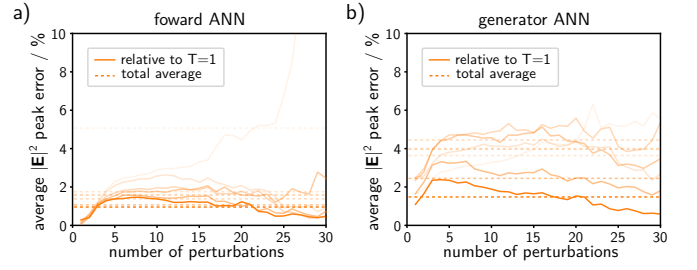


Figure S4. 1×2 MMI (a) forward and (b) generator ANN average absolute output peak intensity error calculated from the validation dataset for each of the iterations. Increasing iterations are indicated by line colour transitioning from light to dark and dashed line shows the total average error.

D. ANN validation statistics and additional inverse design examples

Figures S4a and S4b show the average absolute peak intensity error of the forward and generator ANNs, respectively, for each of the six generations of the etched 1×2 MMI iterative training scheme. The errors are calculated by comparing predicted and simulated outputs of the validation dataset, which remain unseen by the networks during training, and are plotted against the number of perturbations contained by a pattern. The mean total error for any number of perturbations is given by the dashed line and subsequent iterations are indicated by the darker colour of a particular line. The initial generation of the forward network shows very poor prediction with around a 5% average error that rapidly deteriorates with increasing number of perturbations. Later iterations have significantly less error (around 1% for iteration 5) and a much flatter response to number of perturbations. The generator networks show a similar improvement with increasing iterations, however, the error overall is slightly higher as the generator itself also utilises the forward predictor network.

Similarly, the average absolute peak intensity error of the forward and generator ANNs for each of the six generations of the 3×3 MMI iterative training scheme are shown by figures S5a and S5b, respectively. Once again, later iterations are indicated by darker line colours and the average error is compared to the number of perturbations contained by a pattern, but now the error is evaluated for each of the three input channels. Unlike the etched 1×2 MMI case, the forward network for the 3×3 device continues to display a degradation of prediction performance with increasing number of perturbations, even for later iterations. This is attributed to the increased complexity of the system and higher average number of perturbations contained in a pattern. The generator network, however, shows a much flatter response after the first 8 accepted perturbations. Both the forward and generator ANNs show improved performance with later iteration with average errors of less than 4%.

Examples of inverse designed etched 1×2 MMI patterns are shown in figure S6 for different transmittance splitting ratios, smoothly switching the output from channel 1 (top left) to 2 (top right), with a 90% total transmittance target. Patterns

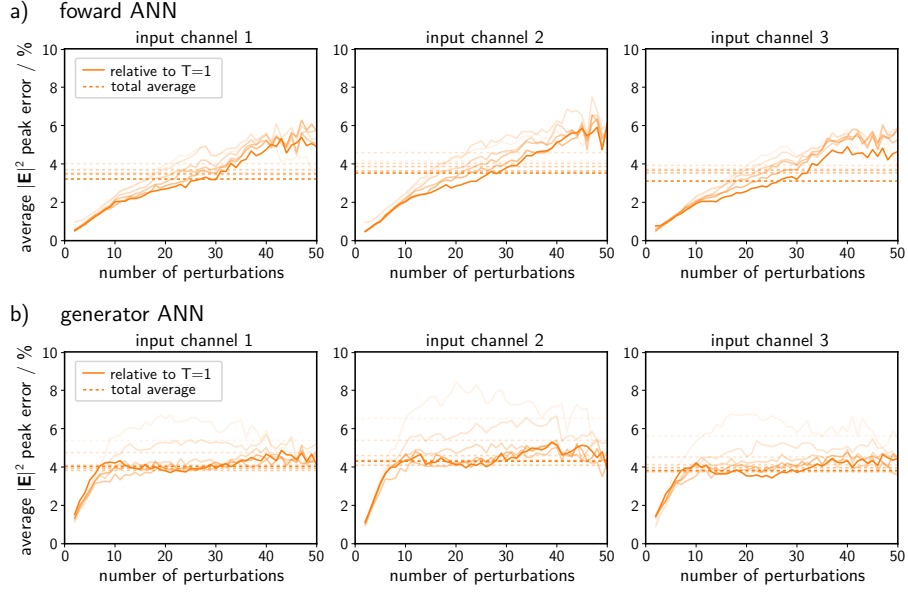


Figure S5. 3×3 MMI (a) forward and (b) generator ANN average absolute output peak intensity error calculated from the validation dataset for each of the iterations and input channels. Increasing iterations are indicated by line colour transitioning from light to dark and dashed line shows the total average error.

were generated by the final iteration of the networks and their corresponding electric field distributions and output intensity profiles were calculated by FDTD simulation. The 45%:45% target case is not shown as it corresponds basically to an empty MMI, the target being very close to the splitting performance of the unperturbed device of 47%:47%. In all cases the agreement between target, prediction and simulation is excellent, as one would expect from performance statistics shown in figure S4. Since the method is data-driven, the network does not necessarily find symmetric patterns for vertically flipped design targets. Note that the right-hand side column cases are identical to the results shown in main text figure 2f and repeated here for ease of comparison.

The remaining two perturbation patterns, and their corresponding electric field distributions, for the complete set of input-output permutations of the 3×3 MMI inverse design examples (figure 4 of the main text) are shown in figure S7a. These have transmittance targets that bare symmetry to previous shown examples, corresponding to flipping the device about the long axis. The complete set of transmission matrices for the port permutation case are then shown in figure S7b, as well as for the additional cases where one port coupling is unitary and the other two are split 50:50 between the remaining outputs. The total transmittance target for each input port was limited to 80% in all cases. The general agreement is very good as shown by the transmission error matrices, however, some specific port couplings have transmittance errors as high as 20%. Note again that the network does not find symmetric patterns for vertically flipped design targets, also the performance of symmetric cases is not identical. This is a result of the model being data-driven.

E. Further phase-aware inverse design examples

In figure S8 we show additional examples for phase-aware inverse design. The color hue and brightness represent the complex phase and amplitude at the output of the device normalized to the input amplitude, as indicated by the color-legend in the figure centre. Subfigures S8a-d represent fixed intensity patterns, for each of which the phase is rotated through a 2π cycle. Fig. S8d shows the same intensity pattern as main text figure 6, with an additional relative phase of $\pi/6$ between the individual output channels.

Table S1. Average number of mismatching perturbations between the inverse designs and the most similar MMI pattern in the training set. The table compares the two different MMI-types and ANN models, trained either only on the initial dataset or via the iterative training. The error corresponds to the standard deviation. The second value is the average number of perturbations of the inverse designed MMIs patterns

MMI type / dataset	most similar training sample avg. mismatch count / total pert. count	
1×2 / initial data only	10.1 ± 10.2	/ 19.1 ± 15.0
1×2 / iterative training	1.9 ± 1.5	/ 16.1 ± 5.1
3×3 / initial data only	21.4 ± 12.5	/ 45.6 ± 20.9
3×3 / iterative training	17.1 ± 12.2	/ 36.0 ± 13.6

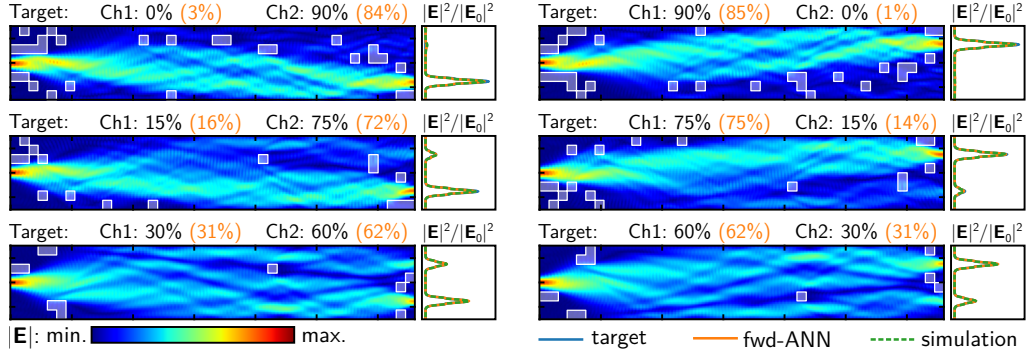


Figure S6. Examples of inverse designed 1×2 MMI perturbation patterns with a 90% total transmittance target. Patterns are generated from the iteration 5 network and their corresponding electric field distributions and output intensity profiles were calculated by FDTD simulation. The target transmittance values for each output port are listed above the electric field distribution (simulated transmission is given in orange), for different splitting ratios switching the output between channel 1 and 2. To the right of the electric field distributions are the intensity profiles along the output waveguides for design target (solid blue lines), forward-ANN (solid orange lines) and FDTD simulation (dashed green lines).

F. Compare inverse design to training data

In order to assess whether the inverse ANN developed a general understanding of the inverse design problem or if it only acts as a kind of lookup table for the training data, we compared each inverse design pattern of the test-set to the full set of training samples. To quantify the “originality” of an inverse design, we search the training set for the most similar sample, hence the pattern with the least number of different

perturbation positions in comparison to the ANN design. In table S1 we show the average number of non-identical perturbation positions and its standard deviation for the generator network trained on either the initial dataset only, or on the dataset after 5 full training iterations. While after the iterative training, the 1×2 generator seems to become some kind of lookup table for the best matching training-set sample (< 2 different perturbation positions), in the 3×3 case, the network does generally not reproduce a pattern from the training set, but produces original MMI designs.

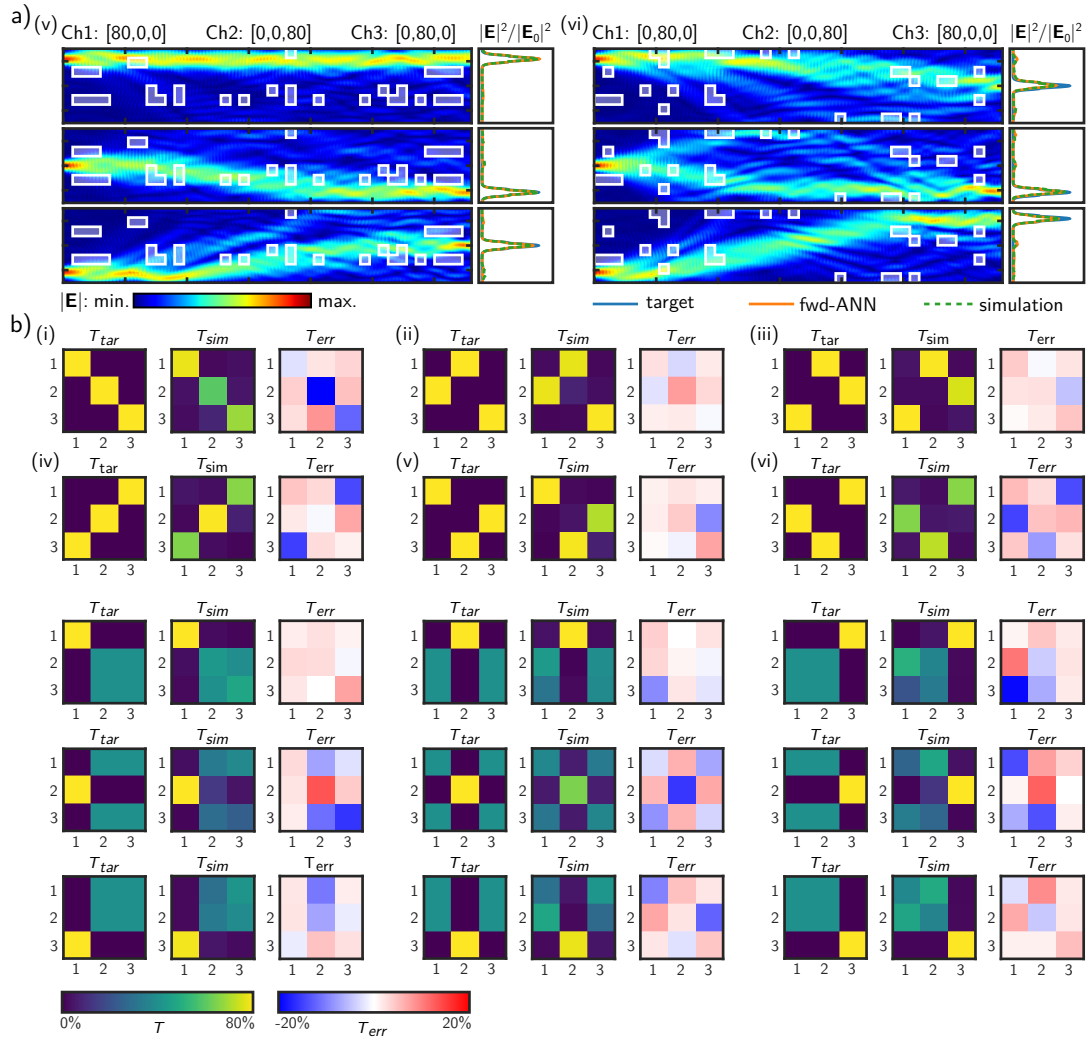


Figure S7. (a) FDTD simulated electric field distributions and output waveguide intensity profiles of the remaining two 3×3 MMI inverse design examples (v-vi) from figure 4 of the main text. Patterns are generated from the iteration 5 network. Transmission matrices for all six input-output permutation cases (roman numerals correspond to part (a) and figure 4 of the main text), as well as for the additional cases where one port coupling is unitary and the other two are split 50:50 between the remaining outputs. The total transmittance target is set to 80%.

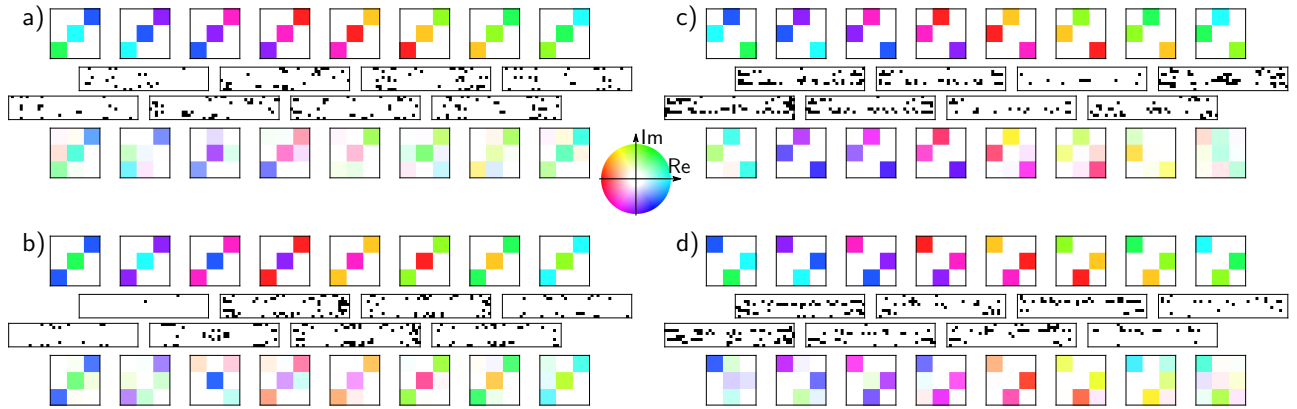


Figure S8. More examples of phase-aware inverse design. a)-d) show a fixed intensity and relative phase layout, for which the phase is rotated through a 2π cycle. Top rows show the complex field target, the center rows the ANN-designed MMI, and the bottom rows the FDTD simulated complex transmission matrix. d) shows the same case as in main text figure 6, but here with an additional relative phase of $\pi/6$ between each non-zero output channel.