Distributed Interference Alignment for K-user Interference Channels via Deep Learning

Rajesh K Mishra*, Karl Chahine*, Hyeji Kim*, Syed Jafar†, and Sriram Vishwanath‡
* University of Texas, Austin
† University of California, Irvine
‡ GenXComm Inc.

2021 IEEE International Symposium on Information Theory (ISIT) | 978-1-5386-8209-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISIT45174.2021.9517735

***Abstract*—In this paper, we develop a framework for an autoencoder based transmission strategy for achieving distributed interference alignment and optimal power allocation in a multi-user interference channel. The users in the interference channel have access to the local channel state information only. We compare the explicit schemes, such as MaxSINR [1], against the autoencoder schemes. We find that the MaxSINR schemes outperform the autoencoder networks which are either jointly or distributively trained from scratch. However, we find that the autoencoders which are pretrained with the beamforming vectors and the power allocation obtained from the explicit schemes outperform the explicit schemes when the interference gets stronger. The explicit schemes perform well as they are effective in choosing the set of users which are to be suppressed. The pretrained autoencoders benefit from this initialization, and also from the fact that end to end training can improve their performance even further. We showcase our performance comparison results for 5 user interference channels with different levels of interference.**

***Index Terms*—Interference channels, sum rate maximization, autoencoders**

I. Introduction

Multiuser communication systems suffer from *interference*, a challenge that requires the careful design of transmission schemes to mitigate said interference. Since first introduced by Ahlswede in 1974 [2], several inner and outer bounds have been derived for interference channels (ICs), establishing the capacity region for special classes of two-user ICs [3], [4], [5]. However, the capacity region for ICs is not known in general for two-user ICs. For ICs with more than two users, very few results are known beyond interference alignment (IA) [6], [7], [8]. IA is a coding scheme that utilizes the fact that each user's signal is corrupted by the combined interference signal in K-user ICs. IA aligns the interfering signal to a subset of the receiver signal space and recovers the desired signal from the orthogonal space. Several IA schemes were proposed (e.g. [9], [10], [11]), and it is also shown that they achieve the optimal degrees of freedom (DoF) of $K/2$ [9] .

From practical perspectives, IA has two limitations; first, they assume that all transmitters and receivers have perfect and global channel knowledge. Second, by focusing on the DoF, IA ignores the strength of each channel and network topology [12]. To resolve these limitations, several distributed coding schemes have been proposed when such channel knowledge is absent [12], [1], [13], [14]. In general, the primary explicit strategies with or without full channel state knowledge are: (a) interference avoidance, where topological properties of the network are used in order to determine the best avoidance scheme, known as topological interference management (TIM), and (b) interference toleration, where interference is present but the transmission scheme is designed by treating interference as noise (TIN). With few exceptions, TIM and TIN make up the majority of explicitly structured transmission schemes with or without channel knowledge at finite signal to noise ratio (SNR).

Deep learning has attained huge interest for communications in general [15], [16], [17], [18], [19], [20], [21], [22] and for interference channels [23], [24]. Centralized autoencoder framework for two-user interference channels was first introduced in [23]. They show that neural network based codes, trained jointly, outperform time-sharing schemes for two-user interference channels. In [24], the authors propose an adaptive deep learning algorithm for K-user symmetric interference channels and show that their algorithm outperforms the conventional system using PSK or QAM. While these results are promising, very important questions are yet to be answered. Does deep learning allow one to devise new interference alignment scheme that outperforms existing IA schemes, especially for asymmetric interference channels? Does deep learning allow us to gain better insight on existing IA schemes? In this paper, we show that the answers to these questions are affirmative. Our main contributions are as follows:

- We introduce an autoencoder framework for distributed interference alignment for K-user interference channels and empirically show that our autoencoder based distributed IA scheme outperforms IA schemes in [1], [12] for asymmetric moderate-to-high interference channels, establishing new state-of-the-art. (Sections III, V-A)
- We introduce a learning methodology that combines deep learning harmoniously with established IA schemes. We empirically find that it is crucial for learning reliable IA schemes (Section III, V-A).
- We run interpretation analysis which shows that the autoencoder-based IA scheme, pre-trained with the beam vectors from explicit algorithm followed by end-to-end training, deviates from the explicit scheme and is able to learn new IA schemes that outperform the original explicit scheme in strong interference regime. By analyzing these results, we can also obtain new insight on improving existing IA schemes (Section V-C).

Authorized licensed use limited to: Access paid by The UC Irvine Libraries. Downloaded on September 02,2021 at 16:47:55 UTC from IEEE Xplore. Restrictions apply.

II. Interference Alignment

We consider a K-user interference channel with K pairs of single-antenna transmitters and receivers comprising of $K(K-1)$ cross channels. As depicted in Figure 1, each transmitter i has a message intended for the corresponding receiver, denoted by $w_i \in \mathcal{N}(0,1)$. The i-th encoder maps w_i to the transmitted symbol vector $X_i \in \mathbb{R}^n$, where n corresponds to a symbol extension over multiple orthogonal time slots. The i-th decoder maps $Y_i \in \mathbb{R}^n$ to $\hat{w}_i \in \mathbb{R}$. The transmitters adhere to the average power constraint $\mathbb{E}\left[\|X_i\|^2\right] \leq 1$ for $i \in [K]$.

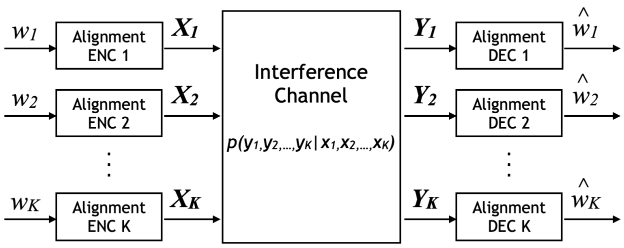

Fig. 1. Interference alignment framework for K-user interference channels. We replace the encoders and decoders by neural networks in Section III.

IA schemes aim to find *linear* encoder and decoder mappings that result in reliable communication, where the reliability is measured by achievable rate or mean square error (MSE). Formally, the encoder is modeled as $X_i = \sqrt{P_i} V_i w_i$, where V_i denotes the beamforming vector and P_i denotes the power allocation. The decoder is modeled as $\hat{X}_i = U_i Y_i$. IA schemes find the set of parameters P_i, V_i, U_i for $i \in [K]$ that optimizes achievable rate or MSE.

A. MaxSINR Algorithm [1]

The authors in [1] propose a distributed optimization of the receive beamforming vectors for coherent combining of the received vectors for Gaussian interference channels, defined as

$$Y_k = \sum_{j=1}^{K} \boldsymbol{H_{kj}} X_j + Z_k \quad \forall k \in [K], \tag{1}$$

where Y_k is the $n \times 1$ received vector, $\boldsymbol{H_{kj}}$ represents the channel coefficient between the j^{th} transmitter and the k^{th} receiver and $Z_k \sim \mathcal{N}(0, N)$. We review MaxSINR algorithm, depicted in Algorithm 1. It makes use of the duality assumption which states that the optimal receive beamformers are also optimal if the receiver is used as a transmitter for the reciprocal channel. The power allocation in the transmitter for the forward communication as $P_k = \mathrm{SNR}^{r_j}$ while that for the reverse direction are defined as $\bar{P}_k = \mathrm{SNR}^{\bar{r}_j}$. These power allocation vectors for the MaxSINR algorithm are computed based on the finding in [12] and shown to hold true for reciprocal channels in [25]:

$$\bar{r}_k = -\max_{j:j\neq k} \{0, \alpha_{kj} + r_j\} \tag{2}$$

$$r_k = -\max_{j:j\neq k} \{0, \alpha_{jk} + \bar{r}_j\} \tag{3}$$

where $P_k = \mathrm{SNR}^{r_k}$ and $\bar{P}_k = \mathrm{SNR}^{\bar{r}_k}$ are the power allocations in the forward direction and the reciprocal directions of the k^{th} transmitter respectively. The computed received beamforming vectors $\boldsymbol{U_k}$ for user k are given as

$$\boldsymbol{U_k} = \frac{\boldsymbol{B_k^{-1} H_{kk} V_k}}{\left\|\boldsymbol{B_k^{-1} H_{kk} V_k}\right\|} \tag{4}$$

where

$$\boldsymbol{B_k} = \sum_{j=1, j\neq k}^{K} P_j \boldsymbol{H_{kj} V_j V_j}^H \boldsymbol{H_{kj}}^H \tag{5}$$

Now we provide a distributed algorithm that was presented in [1] that computes the $\boldsymbol{U_k}$ and the $\boldsymbol{V_k}$ beamforming vectors $\forall k \in [K]$ assuming reciprocity of the interference channel.

Algorithm 1: Distributed MaxSINR

```
for i ≤ K do
    Choose random n × 1 vectors: V_k
    Choose random power allocation: P_k
end
for each iteration do
    for k in K do
        Compute U_k, P̄_k from (5), (4) and (2)
    end
    V̄_k = U_k ∀k ∈ [K]
    for k in K do
        Compute Ū_k, P_k from (4) and (3)
    end
    V_k = Ū_k ∀k ∈ [K]
end
```

B. TIMTIN [25]

TIMTIN algorithm was proposed as a hybrid algorithm to compute the IA beamforming vectors and also the power allocations that are optimal for assuming we treat the interference as noise. Basically, the beamforming vectors are computed to suppress the strong interferers, and the power allocation is done to minimize the effect of the remaining interferers. A distributed solution to the TIMTIN problem was proposed in [25] where the duality property of the interference channel was used to obtain the solution vectors. The algorithm optimizes the minimum interference at the receiver by computing the received beamforming vectors to cancel the strongest interference at the receiver. This is repeated reciprocally at the transmitter end by treating it as the receiver by computing the newly computed received beamforming vectors at the receiver as transmit beamformers. The power allocations vectors are computed as in (2) and (3). For each receiver in the forward channel $\boldsymbol{U_k}$ is computed from the nullspace of $n-1$ strongest interferers that are incident upon the receiver k. These vectors were then used as transmit beamformers in the reciprocal direction and the process was repeated over many iterations to converge to the optimal set of vectors for $\boldsymbol{V_k}$ and $\boldsymbol{U_k}$.

Authorized licensed use limited to: Access paid by The UC Irvine Libraries. Downloaded on September 02,2021 at 16:47:55 UTC from IEEE Xplore. Restrictions apply.

III. Autoencoder for interference channels

The neural encoders and the decoders replace the alignment encoders and decoders to generate the alignment vectors at the transmitter and the receiver along with the power allocation. The autoencoder network for the K-user interference channel was constructed by placing K independent autoencoders trained separately and simultaneously. Each autoencoder neural network has two components, the encoder and the decoder network.

Encoder architecture: The implemented encoder architecture is shown in Fig. 2. The encoder network model is constructed with two fully connected layers with $ReLU$ activation followed by a normalization layer and a scaling layer. Each fully connected layer has 32 nodes. The input vector consists of a single scalar symbol which is projected to a n dimensional power constraint transmit vector at the output of the scaling layer. The encoders of each of the users are separate and do not share any weights between themselves. The fully connected layer implements the function

$$\phi(x) = \sigma\left(W^T x + b\right) \tag{6}$$

where $W \in \mathbb{R}^{L\times N}$ are the parameters of the L neurons of the concerned layer, $b \in \mathbb{R}^L$ is the biasing and $\sigma(\cdot)$ is the activation function.

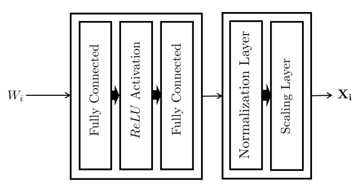

Fig. 2. The architecture of neural encoder i

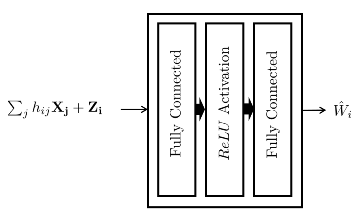

Fig. 3. The architecture of neural decoder i

Power scaling layer: The output of the fully connected layers are normalized through batch normalization in order to have a fixed power constraint. The scaling layer weights each component of the alignment vectors through trainable parameters in order to rearrange the weight of different components of the align. This operation can be represented through the following equation

$$y_i = \frac{\alpha_i}{\sqrt{\sum_i \alpha_i^2}} x_i \tag{7}$$

where x_i and y_i are the componentwise input and output while α_i is the scaling factor for each component of the output vector. In addition, another trainable variable $\in [0,1]$ is introduced for power management across all the users in the network by optimizing the final output from an encoder.

Decoder architecture: The decoder neural network is shown in Fig. 3. It has two fully connected layers with $ReLU$ activation followed by the output layer. Each fully connected layer has 32 nodes. The output layer produces a single scalar as the received symbol. Ideally, the sent and the estimated symbols should be identical. A loss function is used to compute the MSE between the sent and the estimated messages and the weights of the encoders and decoders are trained.

Joint and Distributed training. The neural architecture consists of an encoder and a decoder that are trained through a loss function. The output of the encoder and the targeted value, i.e. the input to the autoencoder, are passed into a loss function in order to compute the MSE. The encoder decoder pair of each user computes an MSE value corresponding to its input and output and the interference from the other pairs. In joint training the MSE from each of these pairs are averaged and are used to train the weight parameters of all the pairs simultaneously. $Adam$ optimizer is used for optimizing the weights based on the gradients computed with the MSE loss function. In case of distributed training, it is ensured that the losses of the users are not shared between themselves. In the process the loss function which is typically the MSE between the input and the output symbols is minimized. The whole operation can be summarized as

$$W^\star = \arg\min_{w_{enc}, w_{dec}} \left\| X - g\left(H\left(f\left(X, w_{enc}\right)\right), w_{dec}\right)\right\|^2 \tag{8}$$

where $W^\star = \{w_{enc}, w_{dec}\}$ are the parameters of the autoencoder for user i. This is separately and simultaneously repeated at all users of the network to get an optimal solution.

In case of K user interference channel, the set of parallel acting autoencoder setup is prepared. Each of the autoencoders tries to solve the equation in (8). The coupling between the users is captured in the channel action where the channel vector and the Gaussian noise are applied on the output from the encoder along with the interference from other users before passing it through as input to the decoder. For training the autoencoder, a set of Gaussian distributed symbols were organized into n_b batches of batch size b_s. A batch size of 2000 was chosen with the number of batches as 2. A learning rate of .001 was used over 200 epochs.

Pretraining: We have discussed that the MaxSINR algorithm has been proven to be optimal in weak interference regime, where treating interference as noise is a reasonable assumption. This is extended to a distributed algorithm discussed in the Algorithm 1. However, the MaxSINR does not perform very well for strong interference regime. In a similar fashion, TIMTIN algorithm was considered to be performing better for strong and hybrid interference regimes but fails to perform in finite SNR regimes, as it focuses on the suppressing interference over maximizing signal to noise and interference ratio (SINR). In a bid to improve the training mechanism for auto encoders and to ensure that we get a better solution, we

Authorized licensed use limited to: Access paid by The UC Irvine Libraries. Downloaded on September 02,2021 at 16:47:55 UTC from IEEE Xplore. Restrictions apply.

use the alignment vector V_k $\forall k \in [K]$ obtained from explicit algorithms to train the encoders of the autoencoder networks; we train the k-th encoder of the autoencoder networks via supervised training with pairs of $(W_k, V_k W_k)$ for $k \in [K]$. We use a training set consisting of 2 batches of 2000 samples and run it for 100 epochs with a learning rate of .001. It is followed by decoder training which is done in the usual way, by minimizing the MSE, and finally we do end to end training to get finer results. As we can see in Figures 4 and 6, pretraining is crucial in learning reliable IA schemes.

IV. Experiment Setup

The channel coefficients h_{ki}, which form the diagonal elements of the channel matrix $\boldsymbol{H_{ki}}$ in (1) are assumed to be scalar constants. We let the noise spectral density of the Gaussian noise to be $N = 1$ and the channel coefficients were chosen according to the different interference parameters.

$$h_{ki} = \begin{cases} \sqrt{\text{SNR}} & \text{if} \quad k = i, \\ (\sqrt{\text{SNR}})^{\alpha_{ki}} & \text{if} \quad k \neq i, \end{cases} \tag{9}$$

where α_{ki} denotes the level of interference.

$$\alpha_{ki} = \begin{cases} 0.9 & \text{w.p.} \quad \beta, \\ 0 & \text{w.p.} \quad 1-\beta, \end{cases} \tag{10}$$

A mixture of strong and weak interference was considered for the interference channel. A series of channels were simulated for each value of β. The interference channel coefficients were set through SNR, α and β parameters. Different intermediate interference regimes were simulated by varying the value of β from 0.5 to 1.0. For simplicity purposes, two different strength of interference parameter are used. The cross channels that have strong interference, α for that channel was chosen to be $\alpha_s = 0.9$ while the the weak interference channels have $\alpha_w = 0$. We specifically implement a 5 user interference channel and obtained the MSE for Gaussian signaling employing various algorithms to mitigate the interference. The parameters that we have implemented for simulation are shown.

Parameter	Value	Type
K	5	fixed
n	2	fixed
α_s	0.9	fixed
α_w	0	fixed
β	0.5 - 1.0	variable

V. Results and Discussion

A. Performance Evaluation

The performance of the algorithms is dependent on β which is the probability of any cross channel having strong interference. The expected number of possible cross channels which have strong interference is given as $\beta K(K-1)$. The users transmit their information through a single dimension and project the interference into the remaining $n-1$ dimensional spaces. Thus, cumulatively the users can effectively cancel a maximum of $K(n-1)$ strong interferers. Thus the number of strong interferers which are not zero-forced is given as

$$\eta = \max\{0, \beta K(K-1) - K(n-1)\} \tag{11}$$

The greater the value of η the greater is the intensity of interference at the receivers. It is also worth noting that the explicit algorithms such as MaxSINR are effective in cancelling the interference for smaller values of η but under perform with higher values. In our simulations, we keep $K = 5$ and $n = 2$ as constant and vary β in order to compare the performance. The performance comparisons between different schemes are done with the MSE in dB between the sent and the estimated symbols as there is a strong correlation between the achievable rate and the MSE between the symbols when the input symbols are Gaussian distributed. We find that the superior performance of the MaxSINR algorithms in weak and intermediate interference regimes could be leveraged to have better performance in strong interference channels through pretraining.

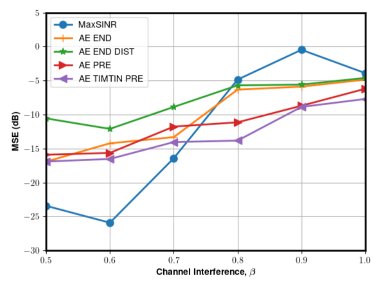

Fig. 4. MSE comparison between the MSE in dB for different algorithm for different interference regimes characterized by the interference parameter β. AE PRE and AE TIMTIN PRE are the autoencoders which are pretrained using MaxSINR and TIMTIN respectively. AE END and AE END represent conventional autoencoder with joint and distributed training respectively

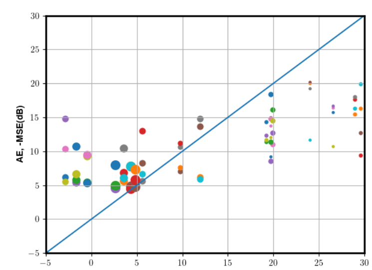

Fig. 5. Different MSE values for all simulated channels with size of the markers defined by the interference parameter β.

Fig. 4 shows the performance of the MaxSINR algorithm compared to other algorithms in different interference regimes. In addition, it also shows the effectiveness of the pretrained algorithms to perform better in strong interference regimes. Specifically, the autoencoders that is trained to the alignment vectors from TIMTIN algorithm shows consistent performance even when compared to a jointly trained autoencoder and a distributed autoencoder. It is seen that the performance

Authorized licensed use limited to: Access paid by The UC Irvine Libraries. Downloaded on September 02,2021 at 16:47:55 UTC from IEEE Xplore. Restrictions apply.

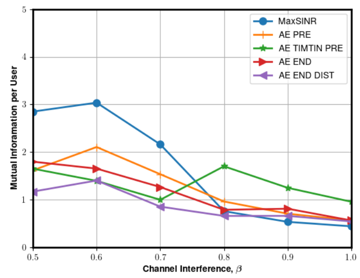

Fig. 6. Mutual Information per user compared across different algorithms for channels with different channel interference parameter β.

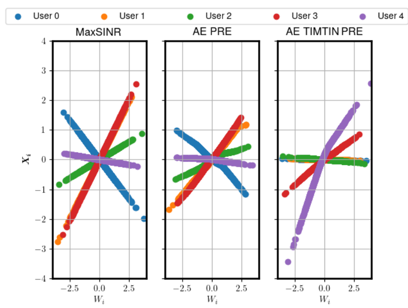

Fig. 7. Constellation Diagram showing the transmitted symbols for different users for the indicated algorithms for a channel with $\beta = 0.8$. Left: MaxSINR, Middle: AE pretrained with MaxSINR. Right: AE pretrained with TIMTIN

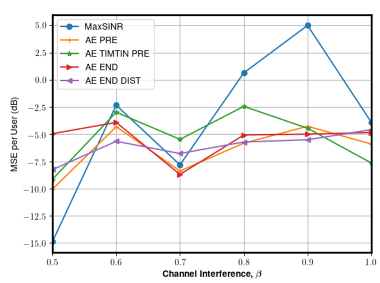

Fig. 8. MSE for different algorithms when they are trained and tested on different channels.

of the MaxSINR for weaker interference regimes is better evident from a lower MSE value but with the increase in β or the amount of interference in the system, MaxSINR starts to underperform. In that respect, the pretraining algorithms although do not match the MaxSINR performance for weaker interference, but at higher interference the average performance in terms of MSE is better. The same behavior is evident in the scatter plots in Fig. 5 where all the channel simulation results are portrayed. The linear plot represents the equal performance. The marker size represents the β value, therefore, higher β implies better performance when compared to the MaxSINR algorithm.

B. Robustness

In order to check the robustness of our networks, a different interference channel matrix was used for training and testing. The strength of the crosschannel was perturbed such that it remained in the same interference regime but its value was changed while the overall channel β was maintained. Fig. 8 plots the MSE when the algorithms were tested for these perturbed channels different than channels that were used for training. When compared to the plots in Fig. 4 we see that the pretrained algorithms are more robust compared to MaxSINR algorithm as the degradation in the MSE is less.

C. Results Interpretation

The superior performance of the MaxSINR algorithm is evident in the weaker interference regimes for smaller values of β. With the increase in the value of β the performance deteriorates as it is not able to effectively cancel all the strong interferers with just $n = 2$ DoF. Fig. 7 shows the output constellation of each of the users. We can see that one of the user is arranged orthogonal to all the other users and thus has the best performance but the performance of all the other users is affected. If we contrast it with the constellation of the pretrained autoencoders, we see that the alignment operation is no more perfectly linear which gives the autoencoders ability to better manage the interference. Therefore, we conclude, pretrained autoencoders show a significant improvement with the increase in the interference as compared to the explicit schemes. We conjecture that the ability of pretrained autoencoders to induce non-linearity in the representation of the symbols helps them improve performance over the explicit schemes through end to end training.

VI. Acknowledgment

This work is supported in part by Samsung. Syed Jafar's work is supported by the grants NSF CNS 1731384, NSF CCF 1907053, ONR N00014-21-1-2386 and ARO W911NF-19-1-0344. Sriram Vishwanath's work is supported by the grants NSF CNS 1731384 and ARO W911NF-18-1-0247.

VII. Conclusions

In this paper, we propose a framework for implicitly developing transmission strategies for multi-user interference channels with finite SNR, where the users only have local channel information. We leverage the effectiveness of the explicit algorithms like MaxSINR and TIMTIN to compute the alignment vectors and the power allocations, and use them to pretrain the autoencoder. We then perform end to end training on top of that to get a better performance when compared to the explicit schemes. We show that with the increase in the degree of interference, the performance of the MaxSINR algorithm deteriorates, but the pretrained autoencoders are able to outperform in these strong interference scenarios. In future work, we intend to extend it to other settings of n with higher number of users. Analyzing the bit error rate (BER) performance with modulated schemes and developing short blocklength codes for interference channels are also left as interesting future work.

Authorized licensed use limited to: Access paid by The UC Irvine Libraries. Downloaded on September 02,2021 at 16:47:55 UTC from IEEE Xplore. Restrictions apply.

References

[1] K. Gomadam, V. R. Cadambe, and S. A. Jafar, "A distributed numerical approach to interference alignment and applications to wireless interference networks," *IEEE Transactions on Information Theory*, vol. 57, no. 6, pp. 3309–3322, 2011.

[2] R. Ahlswede, "The capacity region of a channel with two senders and two receivers," *Ann. Probab.*, vol. 2, no. 5, pp. 805–814, 10 1974. [Online]. Available: https://doi.org/10.1214/aop/1176996549

[3] R. H. Etkin, D. N. C. Tse, and H. Wang, "Gaussian interference channel capacity to within one bit," *IEEE Transactions on Information Theory*, vol. 54, no. 12, pp. 5534–5562, 2008.

[4] C. Nair, L. Xia, and M. Yazdanpanah, "Sub-optimality of han-kobayashi achievable region for interference channels," in *2015 IEEE International Symposium on Information Theory (ISIT)*, 2015, pp. 2416–2420.

[5] M. Costa and A. E. Gamal, "The capacity region of the discrete memoryless interference channel with strong interference (corresp.)," *IEEE Transactions on Information Theory*, vol. 33, no. 5, pp. 710–711, 1987.

[6] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge University Press, 2011.

[7] T. Gou and S. A. Jafar, "Capacity of a class of symmetric simo gaussian interference channels within o(1)," in *2009 IEEE International Symposium on Information Theory*, 2009, pp. 1924–1928.

[8] B. Bandemer, G. Vazquez-Vilar, and A. El Gamal, "On the sum capacity of a class of cyclically symmetric deterministic interference channels," in *2009 IEEE International Symposium on Information Theory*, 2009, pp. 2622–2626.

[9] V. R. Cadambe and S. A. Jafar, "Interference alignment and degrees of freedom of the k-user interference channel," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3425–3441, 2008.

[10] M. A. Maddah-Ali, A. S. Motahari, and A. K. Khandani, "Communication over mimo x channels: Interference alignment, decomposition, and performance analysis," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3457–3470, 2008.

[11] V. R. Cadambe, S. A. Jafar, and S. Shamai, "Interference alignment on the deterministic channel and application to fully connected gaussian interference networks," *IEEE Trans. Inf. Theor.*, vol. 55, no. 1, p. 269–274, Jan. 2009. [Online]. Available: https://doi.org/10.1109/TIT.2008.2008116

[12] C. Geng, N. Naderializadeh, A. S. Avestimehr, and S. A. Jafar, "On the Optimality of Treating Interference as Noise," *IEEE Transactions on Information Theory*, vol. 61, no. 4, pp. 1753–1767, 2015.

[13] D. A. Schmidt, C. Shi, R. A. Berry, M. L. Honig, and W. Utschick, "Minimum mean squared error interference alignment," in *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, 2009, pp. 1106–1110.

[14] S. W. Peters and R. W. Heath, "Interference alignment via alternating minimization," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 2445–2448.

[15] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be'ery, "Deep learning methods for improved decoding of linear codes," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 119–131, 2018.

[16] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Deep learning based MIMO communications," *CoRR*, vol. abs/1707.07980, 2017.

[17] E. Nachmani and L. Wolf, "Hyper-graph-network decoders for block codes," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 2329–2339.

[18] H. Ye, G. Y. Li, B. F. Juang, and K. Sivanesan, "Channel agnostic end-to-end learning based communication systems with conditional gan," in *2018 IEEE Globecom Workshops (GC Wkshps)*, Dec 2018, pp. 1–5.

[19] N. Shlezinger, Y. C. Eldar, N. Farsad, and A. J. Goldsmith, "Viterbinet: Symbol detection using a deep learning based viterbi algorithm," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, July 2019, pp. 1–5.

[20] A. Elkelesh, M. Ebada, S. Cammerer, and S. t. Brink, "Decoder-tailored polar code design using the genetic algorithm," *IEEE Transactions on Communications*, vol. 67, no. 7, pp. 4521–4534, July 2019.

[21] H. Kim, Y. Jiang, S. Kannan, S. Oh, and P. Viswanath, "Deepcode: Feedback codes via deep learning," in *Advances in neural information processing systems*, 2018, pp. 9436–9446.

[22] H. Kim, Y. Jiang, R. Rana, S. Kannan, S. Oh, and P. Viswanath, "Communication algorithms via deep learning," in *International Conference on Learning Representations*, 2018.

[23] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[24] D. Wu, M. Nekovee, and Y. Wang, "Deep learning-based autoencoder for m-user wireless interference channel physical layer design," *IEEE Access*, vol. 8, pp. 174 679–174 691, 2020.

[25] C. Geng, H. Sun, and S. A. Jafar, "Multilevel topological interference management," in *2013 IEEE Information Theory Workshop, ITW 2013*, 2013.

Authorized licensed use limited to: Access paid by The UC Irvine Libraries. Downloaded on September 02,2021 at 16:47:55 UTC from IEEE Xplore. Restrictions apply.