Federated Neural Compression Under Heterogeneous Data

Eric Lei, Hamed Hassani, and Shirin Saeedi Bidokhti
Department of Electrical and Systems Engineering
University of Pennsylvania, Philadelphia, PA
{elei, hassani, saeedi}@seas.upenn.edu

2023 IEEE International Symposium on Information Theory (ISIT) | 978-1-6654-7554-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISIT54713.2023.10206457

***Abstract*—We discuss a federated learned compression problem, where the goal is to learn a compressor from real-world data which is scattered across clients and may be statistically heterogeneous, yet share a common underlying representation. We propose a distributed source model that encompasses both characteristics, and naturally suggests a compressor architecture that uses analysis and synthesis transforms shared by clients. Inspired by personalized federated learning methods, we employ an entropy model that is personalized to each client. This allows for a global latent space to be learned across clients, and personalized entropy models that adapt to the clients' latent distributions. We show empirically that this strategy outperforms solely local methods, which indicates that learned compression also benefits from a shared global representation in statistically heterogeneous federated settings.**

I. Introduction

Traditional learned compression usually takes place in a centralized setting, where one compression model is learned from data collected from various sources and stored at a single location. The standard lossy neural compression models the centralized data as a *single* source $\boldsymbol{x} \sim P_X$ supported on $\mathcal{X}$. Learned compression, under this source assumption, can be set up through the lens of nonlinear transform coding (NTC) [1]. NTC seeks to find an analysis transform $g_a : \mathcal{X} \to \mathcal{Y}$ that maps $\boldsymbol{x}$ to a latent variable $\boldsymbol{y}$, and a synthesis transform $g_s : \mathcal{Y} \to \mathcal{X}$ that maps from the latent space back to the source/reconstruction space. The latent variable $\boldsymbol{y}$ is then quantized to $\hat{\boldsymbol{y}} = \lfloor \boldsymbol{y} \rceil$ such that each entry is quantized to the nearest integer, and entropy coded using likelihoods generated from some entropy model $p_{\hat{\boldsymbol{y}}}(\hat{\boldsymbol{y}})$. The goal is to minimize the operational rate-distortion trade-off

$$\mathbb{E}_{\boldsymbol{x}}[-\log_2 p_{\hat{\boldsymbol{y}}}(\lfloor g_a(\boldsymbol{x}) \rceil)] + \lambda \, \mathbb{E}_{\boldsymbol{x}}[\mathsf{d}(\boldsymbol{x}, g_s(\lfloor g_a(\boldsymbol{x}) \rceil))], \qquad (1)$$

where $\lambda > 0$ controls the trade-off. The transform functions g_a, g_s, as well as the entropy model are all parameterized using neural networks.

In practice, NTC models are typically trained on aggregated datasets such as MS-COCO [2] or ImageNet [3], which are collected from many clients into one location and form the samples from the source $\boldsymbol{x}$. Then, a single model is trained on this source using the objective in (1). This centralized approach has been successful in a wide variety of applications and modalities such as image compression [4]–[6], compression of channel state information (CSI) in wireless communications [7], [8], and in audio compression [9].

However, in many cases, a centralized approach is not necessarily feasible or applicable to the end user's setting. Rather than already existing in a central location, the data may be scattered across clients and it may not be feasible to centrally collect it due to limitations such as privacy constraints. Moreover, it can be expensive to collect all data at a centralized location when there are many clients. These challenges motivate analyzing learned compression from a distributed setup where multiple clients all wish to learn a compressor for their respective data in a federated fashion, potentially with the help of a central server.

One immediate challenge that arises is that there are now potentially n *different* sources, where n is the number of clients. The single source NTC setup in (1) is, however, formulated for a single source. This parallels a well-known challenge in federated learning (FL), which is a related problem where clients wish to jointly learn a classifier or regressor from their data, where the data across clients are statistically heterogeneous, yet may share some underlying structure. For example, medical images collected using different equipment or in different locations may be modeled by different source distributions; yet, they are all fundamentally the same type of image. In the context of data compression, a natural question arises: how to model common structure across statistically heterogeneous sources? Furthermore, how can this shared structure be leveraged by the clients in order to learn good lossy compressors for each client?

For the former question, one difference that precludes direct application of the FL setup is that in FL, the statistical heterogeneity is modeled within the labels of the data distribution, not the features (e.g., pixel values in images). Learned compression is, however, an unsupervised task, and has no labeled data. Thus, all statistical heterogeneity and common structure that may exist across client data needs to be modeled solely within the features. In our work, we take these considerations into account by modelling the clients' sources as distributions induced by a shared generative function f applied to n independent latent sources. This shared generative function f ensures that all sources share a common feature space.

For the latter question, a naive approach to solve this problem is to try to learn a single NTC-based compressor, where clients send model updates to the server at each round, and the server computes the average of the received models.

This approach would extend FedAvg [10] in FL to this federated compression setting. However, under our source heterogeneity assumption, it is more natural to learn client-specific compressors that are better tuned to each client's source distribution. One client-specific solution is to train a local model for each client. A challenge this approach faces in practice is that the number of samples each client has is small relative to the number of clients. Under our data heterogeneity model, however, there is intrinsic structure in the feature space defined by the shared function f, which can be leveraged across client data. We thus propose a novel solution with NTC models: learn globally shared analysis and synthesis transforms, with client-specific entropy models. This approach allows the shared analysis and synthesis transforms to try and undo the shared function f, extracting the client-specific latent source, which is then compressed using client-specific entropy models.

Our contributions are as follows.

1) We propose a federated learned compression framework that simultaneously encompasses statistical heterogeneity and shared intrinsic structure across clients.
2) We propose a novel method to learn client-specific neural compressors that leverages the shared structure across clients.
3) We demonstrate empirically that learning a shared analysis and synthesis transform followed by locally optimized entropy models performs better than solely local NTC models.

II. Related Work

A. Neural Compression

The use of neural networks to design lossy compressors was initiated by merging quantization with autoencoder architectures [11]–[14]. These methods, under the umbrella of NTC [1], operate by mapping the source to a latent space and back by using an analysis and synthesis transform parameterized with neural networks. Uniform quantization and entropy coding (using an additional learned entropy model) is then performed in the latent space, which is generally much lower in dimensionality. More recent work improve upon the entropy modeling to remove additional redundancy in the latent space. The scale hyperprior [4] transmits side information to generate different entropy models for each sample, with [5] and [6] building on top of this via autoregression of the latent variable. In single-image compression, these methods have shown to outperform recent handcrafted codecs such as HEVC [15] and VVC [16].

These methods are typically trained on data that has been pre-collected and stored centrally. One model is typically trained and deployed. In contrast, our work considers the case where the data resides across distributed clients such that collecting them centrally is not possible, and designing architectures to learn good compressors in such a setting.

B. Federated Learning

There have been many recent works discussing federated learning (FL), especially under client heterogeneity [17]–[20]. Most of these works have been applied in classification or regression settings, where each client i has its own training data $\boldsymbol{x}, \boldsymbol{y} \sim P_i$ and wishes to learn a good predictor that generalizes well on their respective distribution P_i. One popular approach is FedAvg [10], where a single model is sent to clients to update the model, before the central server averages their updates. Another approach is FedRep [17], where a global feature extractor is learned on centralized data, and local client heads are jointly trained which use the extracted features to predict on local client data. Both methods have been shown to recover the underlying shared representation across clients in simple regression settings [21].

One difference in our setup is that learned compression does not assume labeled data. The assumptions of personalized FL are that heterogeneity resides in the labels, and hence the natural architecture to arise is a shared feature extractor followed by a predictor that is personalized. In contrast, we propose a different notion of heterogeneity with shared structure that purely resides in the features. We argue that this naturally leads to an architecture with shared analysis and synthesis transforms followed by personalized entropy model.

C. Distributed Neural Compression

There have also been recent works in designing distributed neural compression schemes [22]–[25]. These works are inspired by the information-theoretic results on compression with side information [26], [27], which say that if one has side information on a source to be compressed, an encoder that does not observe side information can perform just as well as one that does (in both cases the decoder observes the side information). However, this is slightly different from the federated setting. In contrast, we are interested in learning n point-to-point lossy compressors (i.e., one for each client), where each compressor does not assume any side information from the other clients to be available. Rather than exploiting side information (i.e., a correlated source), we want to exploit underlying structure of the n sources, despite being statistically independent.

III. Problem Formulation

A. Single-Source Neural Compression

As discussed in the introduction, classical neural compression models the data as a single source $\boldsymbol{x} \sim P_X$, and attempts to learn a model that performs well in terms of rate and distortion defined w.r.t. $\boldsymbol{x}$. The main intuition is that sources such as images have a low-dimensional latent space which can be extracted by the analysis and synthesis transforms. The low-dimensional latent representation is modelled probabilistically using a learned entropy model, which quantizes and entropy codes the latent variable. A figure describing this setup is shown in Fig. 1.

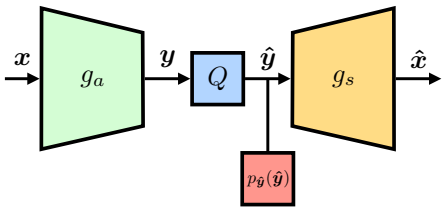


Fig. 1: Single source neural compression.

Algorithm 1 Local-NTC

Require: Total iterations T, step size η, rate-distortion trade-off $\lambda > 0$
Initialize $\{g_a^{(i)}\}_{i=1}^n, \{g_s^{(i)}\}_{i=1}^n, \{p_{\hat{\boldsymbol{y}}}^{(i)}\}_{i=1}^n$
for $t = 1, \dots, T$ **do**
  for each client i **do**
    $(g_a^{(i)}, g_s^{(i)}, p_{\hat{\boldsymbol{y}}}^{(i)}) \leftarrow (g_a^{(i)}, g_s^{(i)}, p_{\hat{\boldsymbol{y}}}^{(i)}) - \eta \nabla_{(g_a^{(i)}, g_s^{(i)}, p_{\hat{\boldsymbol{y}}}^{(i)})} (R_n + \lambda D_n)$
  end for
end for

This method encompasses centralized training, where datasets are aggregated ahead of time and a single model is trained using the combined data.

B. Federated Compression

In the federated setting, instead of a single, effectively centralized source, we have n sources $P_1, \dots, P_n$, with $\boldsymbol{x}^{(i)} \sim P_i$ consisting of the data seen by client i. Centralized training corresponds to aggregating all sources into the single source setup with $P_X = \frac{1}{n}\sum_{i=1}^n P_i$. Typically, however, since the sources are heterogeneous across clients, it is better to have client-optimized models. Thus, the most general federated compression setup consists of n separate compressors, such that each client i has its own analysis and synthesis transforms $g_a^{(i)}$, $g_s^{(i)}$, as well as entropy model $p_{\hat{\boldsymbol{y}}}^{(i)}$. Thus, the local federated objective consists of

$$\min_{\{g_a^{(i)}, g_s^{(i)}, p_{\hat{\boldsymbol{y}}}^{(i)}\}_{i=1}^n} \frac{1}{n}\sum_{i=1}^n \big(\mathbb{E}_{\boldsymbol{x}^{(i)}}[-\log_2 p_{\hat{\boldsymbol{y}}}^{(i)}(\lfloor g_a^{(i)}(\boldsymbol{x}^{(i)})\rceil)] + \lambda\, \mathbb{E}_{\boldsymbol{x}^{(i)}}[\mathsf{d}(\boldsymbol{x}^{(i)}, g_s^{(i)}(\lfloor g_a^{(i)}(\boldsymbol{x}^{(i)})\rceil))]\big), \quad (2)$$

where the goal is minimize the rate and distortion averaged over the clients. We call the objective in (2) Local-NTC, whose training algorithm can be found in Alg 1.

Rather than learn a local model for each client, results from federated learning suggest that heterogeneous data can potentially benefit from first learning a shared common representation. In the FedRep framework [17], a common representation is extracted from a shared feature extractor, before individual client heads are learned to predict from this fixed representation. The intuition behind these works is the assumption that data heterogeneity exists primarily in the labels of the dataset, whereas the data samples (e.g., covariates or features) have a shared common structure. Learned compression, on the other hand, is a fully unsupervised task, and so both the heterogeneity and shared structure need to be modelled within the features.

C. Heterogeneous Source Modelling

As an example, if one wishes to compress similar types of images (e.g., medical images) that are collected using different equipment across the clients, one might expect that these images lie on the same image manifold despite the image distributions being different across clients. Mathematically, we can model $P_1, \dots, P_n$ as follows. Let $\{\boldsymbol{z}^{(i)}\}_{i=1}^n$ be the latent sources of randomness underlying the images sources, which could represents properties of the images such as class, orientation, lightning, or style. Then, the sources in the ambient space which the clients observe are generated as $\boldsymbol{x}^{(i)} = f(\boldsymbol{z}^{(i)})$, for some fixed but unknown function f, which models the shared source manifold, but the statistically heterogeneity is modelled implicitly via the $\boldsymbol{z}^{(i)}$'s.

In Local-NTC, the analysis and synthesis transforms at each client attempt to "undo" this generative function f in order to recover $\boldsymbol{z}^{(i)}$ and model its probability density. However, each client may be data limited, and thus learning f at every client's model may be difficult to accomplish. Instead, in order for each client i to recover the underlying source, it may be easier to globally learn the function f across clients. The NTC framework itself naturally possesses an architecture that can naturally support this heterogeneous source model, where the images across clients share an underlying latent representation.

D. Sharing Analysis and Synthesis Transforms

The analysis transform can be seen as a feature extractor that extracts a latent representation (typically of lower dimensionality), which is then quantized and entropy encoded according to a learned entropy model. One can thus view the "prediction head" of learned compression as the entropy coding part of the model. At the decoder side, the decompressed latent variable is transformed back to the reconstruction using the synthesis transform. Thus, if one learns a shared analysis and synthesis transform g_a, g_s across clients, the induced distribution of the latent variables for client i will be $g_a(\boldsymbol{x}^{(i)})$. A client-specific entropy model $p_{\hat{\boldsymbol{y}}}^{(i)}$ is then fine-tuned individually to more accurately model the distribution induced by $g_a(\boldsymbol{x}^{(i)})$. Similar to federated learning, we expect this scheme to perform better than solely local training when the data per-client is small; this helps the analysis and synthesis transforms learn better feature extractors by leveraging all the data across clients. We call this setting Fed-NTC, where the objective is

$$\min_{g_a, g_s, \{p_{\hat{\boldsymbol{y}}}^{(i)}\}_{i=1}^n} \frac{1}{n}\sum_{i=1}^n \big(\mathbb{E}_{\boldsymbol{x}^{(i)}}[-\log_2 p_{\hat{\boldsymbol{y}}}^{(i)}(\lfloor g_a(\boldsymbol{x}^{(i)})\rceil)] + \lambda\, \mathbb{E}_{\boldsymbol{x}^{(i)}}[\mathsf{d}(\boldsymbol{x}^{(i)}, g_s(\lfloor g_a(\boldsymbol{x}^{(i)})\rceil))]\big), \quad (3)$$

which jointly optimizes the client-averaged rate and distortion $R_n + \lambda D_n$, where R_n and D_n are defined as

$$R_n := \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\boldsymbol{x}^{(i)}}[-\log_2 p_{\hat{\boldsymbol{y}}}^{(i)}(\lfloor g_a(\boldsymbol{x}^{(i)})\rceil)], \quad (4)$$

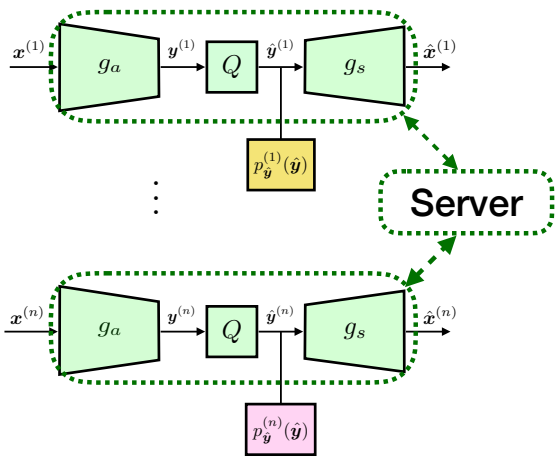


Fig. 2: Federated neural compression under Fed-NTC framework. Analysis and synthesis transforms are learned globally through a central server, whereas the entropy models are learned per-client.

Algorithm 2 Fed-NTC

Require: Participation rate r, communication rounds T, entropy model updates T_p, transform updates T_g, step size η, rate-distortion tradeoff $\lambda > 0$

Initialize g_a, g_s, $\{p_{\hat{\boldsymbol{y}}}^{(i)}\}_{i=1}^n$

for $t = 1, \dots, T$ **do**

Server sends g_a, g_s to random fraction r of clients I_r

for T_p iterations **do**

$p_{\hat{\boldsymbol{y}}}^{(i)} \leftarrow p_{\hat{\boldsymbol{y}}}^{(i)} - \eta \nabla_{p_{\hat{\boldsymbol{y}}}^{(i)}} (R_n + \lambda D_n), \quad \forall i \in I_r$

end for

Initialize $(g_a^{(i)}, g_s^{(i)}) \leftarrow (g_a, g_s), \forall i \in I_r$

for T_g iterations **do**

$(g_a^{(i)}, g_s^{(i)}) \leftarrow (g_a^{(i)}, g_s^{(i)}) - \eta \nabla_{(g_a^{(i)}, g_s^{(i)})} (R_n + \lambda D_n)$

end for

Each client i sends $(g_a^{(i)}, g_s^{(i)})$ back to server

$(g_a, g_s) \leftarrow \frac{1}{rn} \sum_{i \in I_r} (g_a^{(i)}, g_s^{(i)})$

end for

$$D_n := \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\boldsymbol{x}^{(i)}} [\mathsf{d}(\boldsymbol{x}^{(i)}, g_s(\lfloor g_a(\boldsymbol{x}^{(i)}) \rceil))]. \tag{5}$$

Fig. 2 describes the training framework. To optimize (3), one can alternate global updates, which are coordinated through the use of a server as shown in Fig. 2, with local entropy model updates. The full algorithm is shown in Alg. 2.

E. Rate-Distortion Function For Heterogeneous Sources

In this section, we explain why sharing analysis and synthesis transforms is principled, from a rate-distortion point of view. In general, the rate-distortion function of n independent (but not necessarily identical) sources $\boldsymbol{x}^{(i)} \sim P_i$ is achievable using a locally optimal code for each source, with an appropriate rate allocation across the n sources. As such, the optimal trade-off between client-averaged rate and distortion $R^{\mathsf{fed}}(D)$ is given by [28, Sec. 10.3.3]

$$R^{\mathsf{fed}}(D) = \min_{D_1, \dots, D_n : \frac{1}{n} \sum_{i=1}^n D_i \le D} \frac{1}{n} \sum_{i=1}^{n} R_i(D_i), \tag{6}$$

where $R_i(D)$ is the rate-distortion function of P_i. This result appears to suggest that to obtain the best rate-distortion trade-off, it suffices to should learn n separate compressors, one for each client, which is what Local-NTC does. However, under our proposed heterogeneous source model described in Sec. III-C, codes that share a global transform can also achieve $R^{\mathsf{fed}}(D)$ using the following structure. Setting $g_a = f^{-1}$ and $g_s = f$, first transform $\boldsymbol{x}^{(i)}$ to $\boldsymbol{z}^{(i)}$, which is compressed using an optimal code with respect to distortion function

$$\mathsf{d}_z(\boldsymbol{z}^{(i)}, \hat{\boldsymbol{z}}^{(i)}) := \mathsf{d}(f(\boldsymbol{z}^{(i)}), f(\hat{\boldsymbol{z}}^{(i)})). \tag{7}$$

The compressed version of $\boldsymbol{z}^{(i)}$ is transformed back using f. Thus, assuming that $\boldsymbol{z}^{(i)}$ can be optimally compressed, both Local-NTC and Fed-NTC possess the architectures to perform optimal compression using this above scheme. However, in Local-NTC, each user's pair of transforms will need to learn f^{-1} and f individually, whereas in Fed-NTC, f only needs to be learned for a single set of transforms at the global level, which should benefit learning-based compressors, especially when samples are limited. An algorithmic analysis of Fed-NTC is left for future work.

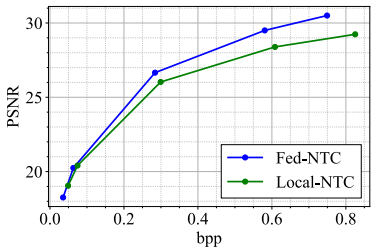
IV. Experiments

We first discuss how we set up the experiments in order to introduce heterogeneity in the data, followed by their results.

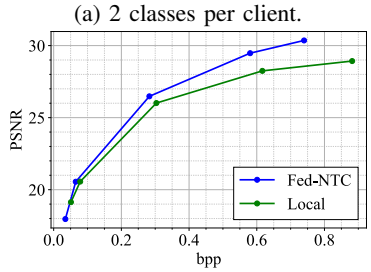
A. Experimental Setup

1) Datasets: To experimentally test the Fed-NTC framework, we test a federated setup on image compression of SVHN, and CIFAR10 datasets, which are all 32×32 RGB image datasets. While generally used for classification purposes, we use their class identities to introduce data heterogeneity across the clients. SVHN and CIFAR10 are both 10-class datasets. We assign S classes to each client, on average, by using the non-i.i.d. data partitioning method detailed in [10]. We then vary S in order to vary the heterogeneity across clients, where $S \in \{2, 5\}$. The number of training samples per user is fixed at N/n, where N is the total number of training samples. This way of introducing heterogeneity can be viewed as the latent source $\boldsymbol{z}^{(i)}$ representing features of the subset of classes; for example, different poses and styles of the class object within the image frame. The generative mapping f then projects these features to the image manifold.

2) Compression Models: For the compression models, we use the NTC methods detailed in [1], with a factorized prior for $p_{\hat{\boldsymbol{y}}}$, which models each entry of $p_{\hat{\boldsymbol{y}}}$ independently using a single-variable density model parameterized by a neural network. In practice, all spatial elements along each channel dimension are modeled by the same density model in order to maintain translation-invariance across the entire model [4]. We add uniform noise to the $g_a(\boldsymbol{x})$ to serve as a proxy during

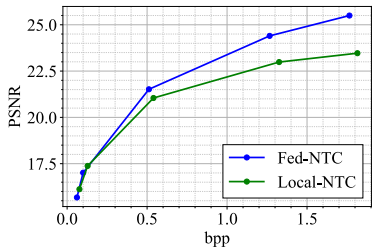


(a) 2 classes per client.

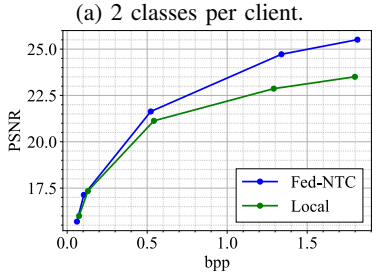


(b) 5 classes per client.

Fig. 3: SVHN with 100 clients.



(a) 2 classes per client.



(b) 5 classes per client.

Fig. 4: CIFAR10 with 100 clients.

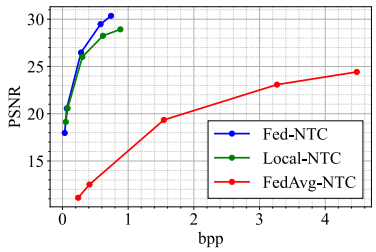
training, with hard quantization and entropy coding during evaluation of the models.

3) Federated Setup: For the federated setup, we use a total of $n = 100$ clients in all image compression experiments. For the baseline model (Local-NTC), which consists of each client having its own model, each client trains its model on its own local data, e.g. in Alg. 1, with no communication among the clients. For the proposed Fed-NTC method, we use a participation rate of $r = 0.1$ in Alg. 2 and a total of 100 rounds of communication. For evaluation, the number of bits needed to compress the hard-quantized latents, and the distortion are measured and averaged for the final 10 rounds of communications.

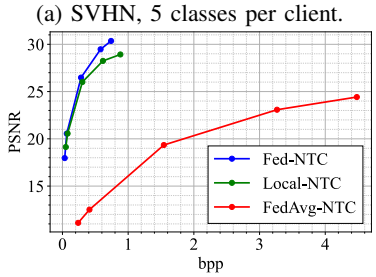
B. Results

In Fig. 4 and 3, we plot the comparisons of Fed-NTC with Local-NTC for CIFAR10 and SVHN, respectively. We observe that in almost all settings, Fed-NTC outperforms Local-NTC models. This indicates that learning a shared representation is able to compress the data across clients better at a variety of different levels of heterogeneity, and that shared generative function is a plausible model for the heterogeneous data with shared structure. The Fed-NTC models are able to recover the global functon f by leveraging the data across clients.

We additionally compare Fed-NTC with a model that shared a single global model across all users. This model, trained under the FedAvg scheme [10], alternates between 10 local updates for each client, before sending back to the server which averages all models. As shown in Figs. 5, such a model performs significantly worse than Fed-NTC and even local training, indicating that a single global model struggles with client heterogeneity in learned compression. We noted no significant difference if the number of local updates was reduced in an effort to reduce client drift. Further experimental results can be found in the extended version of this paper.



(a) SVHN, 5 classes per client.



(b) CIFAR10, 5 classes per client.

Fig. 5: Comparison with a single global NTC model trained under FedAvg.

V. Conclusion

We propose a federated framework for learned compression, where the data across clients may be heterogeneous. We propose a model for such data that has shared structure in the form of a common generative function. This model suggests that in compression, one should first extract a common latent space before client-specific entropy modelling takes place. Experimental results confirm that this assumption is true across a broad class of settings and this scheme is superior to solely local models. Potential avenues for future work include analyzing privacy aspects and algorithmic analysis of different federated schemes.

REFERENCES

[1] J. Ballé, P. A. Chou, D. Minnen, S. Singh, N. Johnston, E. Agustsson, S. J. Hwang, and G. Toderici, "Nonlinear transform coding," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 2, pp. 339–353, 2021.

[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[4] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436*, 2018.

[5] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," *Advances in neural information processing systems*, vol. 31, 2018.

[6] D. Minnen and S. Singh, "Channel-wise autoregressive entropy models for learned image compression," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 3339–3343.

[7] Q. Yang, M. B. Mashhadi, and D. Gündüz, "Deep convolutional compression for massive mimo csi feedback," in *2019 IEEE 29th international workshop on machine learning for signal processing (MLSP)*. IEEE, 2019, pp. 1–6.

[8] S. Ravula and S. Jain, "Deep autoencoder-based massive mimo csi feedback with quantization and entropy coding," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6.

[9] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[10] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[11] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, "Variable rate image compression with recurrent neural networks," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://arxiv.org/abs/1511.06085

[12] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=rJxdQ3jeg

[13] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=rJiNwv9gg

[14] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. V. Gool, "Soft-to-hard vector quantization for end-to-end learning compressible representations," *Advances in neural information processing systems*, vol. 30, 2017.

[15] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[16] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the versatile video coding (vvc) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.

[17] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 2089–2099.

[18] A. Mitra, R. Jaafar, G. J. Pappas, and H. Hassani, "Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients," *Advances in Neural Information Processing Systems*, vol. 34, pp. 14 606–14 619, 2021.

[19] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5132–5143.

[20] A. Reisizadeh, I. Tziotis, H. Hassani, A. Mokhtari, and R. Pedarsani, "Straggler-resilient federated learning: Leveraging the interplay between statistical accuracy and system heterogeneity," *IEEE Journal on Selected Areas in Information Theory*, vol. 3, no. 2, pp. 197–205, 2022.

[21] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Fedavg with fine tuning: Local updates lead to representation learning," *arXiv preprint arXiv:2205.13692*, 2022.

[22] M. B. Mashhadi, Q. Yang, and D. Gündüz, "Distributed deep convolutional compression for massive mimo csi feedback," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2621–2633, 2021.

[23] N. Mital, E. Özyılkan, A. Garjani, and D. Gündüz, "Neural distributed image compression using common information," in *2022 Data Compression Conference (DCC)*. IEEE, 2022, pp. 182–191.

[24] N. Mital, E. Özyilkan, A. Garjani, and D. Gündüz, "Neural distributed image compression with cross-attention feature alignment," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 2498–2507.

[25] J. Whang, A. Acharya, H. Kim, and A. G. Dimakis, "Neural distributed source coding," *arXiv preprint arXiv:2106.02797*, 2021.

[26] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[27] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, 1976.

[28] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. USA: Wiley-Interscience, 2006.

ACKNOWLEDGMENTS

The work of Eric Lei was supported by a NSF Graduate Research Fellowship. The work of Shirin Saeedi Bidokhti was supported by NSF award 1910594 and an NSF CAREER award 2047482. The work of Hamed Hassani was supported by NSF award CIF-1943064. This work was also partially supported by the AI Institute for Learning-Enabled Optimization at Scale (TILOS), NSF award CCF-2112665.