A Compression Algorithm for Distributed LMMs with Different Information Fusion Techniques

Eren Balevi
Qualcomm Inc.
erenbalevi@gmail.com

Abstract

A typical large multimodal model (LMM) involves several encoders, one for each modality, for contextual encoding. Transmission of the encoder outputs, potentially residing at different devices, can lead to significant and intolerable communication overhead for resource-constrained environments. Then, a large language model (LLM) combines the encoded sources with text that can be regarded as side information before the generative process. This structure resembles the Wyner-Ziv problem that promises considerable compression of multiple correlated sources. Motivated by the Wyner-Ziv theorem, we propose a novel compression algorithm for the encoded sources and examine it in terms of semantic efficiency. The developed algorithm is applied to two architectures in terms of performance-complexity tradeoff, namely incorporation of sources (i) at the beginning (for best performance) and (ii) at the later layers (for fast inference) of a decoder. The results indicate that the compression for fast inference has less impact on bad (noisy/low-throughput) channels than on the best performance case, and the semantic similarity can be moderately preserved under certain circumstances. Additionally, the performance drop is negligible for certain compression ratios in both approaches.

1 Introduction

In a multi-user environment, compression can be greatly boosted by the processing of correlated sources in different devices. This is particularly true when multiple users possess highly correlated input sources due to their proximity and shared environment. This has been well proven with the theoretical results of Slepian and Wolf [1973], and Wyner and Ziv [1976]. In this paper, motivated by the inherent distributed nature, the compression limits of large multimodal models (LMMs) are investigated through the lens of the Wyner-Ziv problem.

The seminal work of Slepian-Wolf proves that the case of side information available to both encoder and decoder can achieve the same lossless compression rate when the encoder does not access the side information if the joint distribution of the sources is known Slepian and Wolf [1973]. As a generalization to lossy compression, the same problem was studied when the decoder accesses the correlated side information directly by Wyner and Ziv [1976]. Despite the motivating and surprising results, there have been limited structured coding schemes that establish constructive practical frameworks. One reason for this is that the random coding argument, which divides the set of jointly typical sequences into bins in the proof of distributed coding, does not specify how practical binning should be implemented.

Most of the existing constructive coding studies are limited to special distribution and correlation structures, e.g., Gaussian or doubly symmetric binary sources, and there is a lack of a general framework for any distribution and correlation. Data-driven methods, mainly referred to as neural

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI and ML for Next-Generation Wireless Communications and Networking (AI4NextG).

coding in the literature, can promise the generalization of distributed coding to arbitrary correlation scenarios and input types. In this direction, neural coding has been studied in the context of distributed source coding by Whang et al. [2024], Ozyilkan et al. [2024]. Specifically, Whang et al. [2024] parametrizes an encoder-decoder pair as a vector-quantized variational autoencoder (VQ-VAE) and trains it for the distributed compression of arbitrarily correlated sources. VQ-VAE learns a codebook in the process to be utilized for the quantization of the latents. Ozyilkan et al. [2024] proposes an unstructured entropy-constrained vector quantization (ECVQ) that utilizes side information without imposing any particular structure. They show that a learned compressor exhibits the highly used binning mechanisms in information theory as well as the optimal combination of the quantization index and side information.

Foundation models are utilized within the scope of neural coding as well. Being excelled at the prediction of next tokens, foundation models can unlock efficient compression algorithms relying on the interplay between prediction and compression in information theory. To illustrate, Valmeekam et al. [2023] suggests to exploit a pre-trained large language model (LLM) for lossless compression by combining the output of LLMs (or the probabilities of tokens) with a compressor, such as arithmetic coding and zlib. In the same manner, competitive compression rates are presented for different data modalities with the lossless compression capabilities of foundation models in Deletang et al. [2024]. These promising initial studies do not consider a distributed setting.

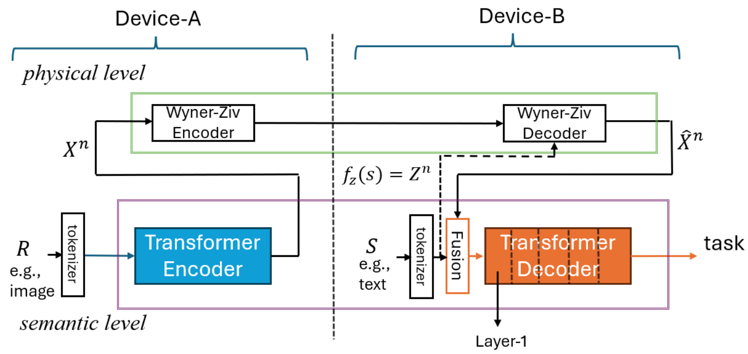
This paper considers a distributed scenario by integrating LMMs with Wyner-Ziv coding. Specifically, a novel compression algorithm for arbitrary distributions is developed for the LMM encoder built upon Wyner-Ziv coding. The proposed algorithm is applied to the two fundamental information-combining techniques. The first one combines the sources at the front-end of a decoder to focus on the performance at the expense of more computational complexity, termed as *information-theoretic or best performance approach*. The second one postpones combining to later decoder layers to speed up inference by avoiding the computation cost of some layers, named as *fast-inference approach*. Our algorithm can make a significant communication overhead saving of around $\%90$ possible. The developed algorithm is experimented with one of the state-of-the-art LMMs. Precisely, LLaVa architecture is adapted to our problem formulation, and the efficiency of the compression is assessed through multiple metrics with precision, recall, and BERTScore Zhang et al. [2020]. Our empirical results showcase that even a large compression ratio does not lead to a significant semantic loss.

The paper is organized as follows. Section 2 introduces the problem statement. A compression algorithm is proposed in Section 3 with the simulation results in Section 4. The paper ends with the concluding remarks 5.

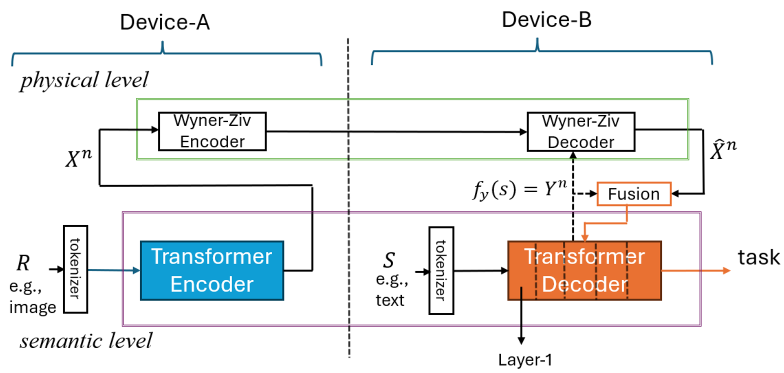
2 Problem Statement

LMMs typically employ a separate encoder for each input modality and therefore are composed of several transformer encoders and a transformer decoder that combines the multi-modal encoder outputs with the input prompt, e.g., Grattafiori et al. [2024], Liu et al. [2023]. The generic structure of LMMs resembles the canonical Wyner-Ziv problem, where the input sources, e.g., image and speech, are decoded with the help of side information, e.g., text. However, the existing LMMs are not designed according to the Wyner-Ziv theorem. This can lead to extra communication overhead for split inference. Despite the growing popularity of LMMs being leveraged in various research applications across multiple scopes, managing and optimizing inference in terms of communication and computation constraints remains an open problem, as it depends on many factors that can change dynamically and is an actively studied research problem.

A key factor for distributed inference that affects compression ratio and distortion (or more generally, performance) is how to combine the encoded inputs while decoding. Several techniques are observed in the design of LMMs to incorporate multiple inputs in the decoder. We classify and name them as “*information-theoretic/best performance approach*" and “*fast-inference approach*". The former integrates the input at the front of the transformer decoder without being processed by the transformer layers and avoids any information loss due to data processing inequality from the information theory perspective. The main cost of this method is the increased complexity of inference. The fast-inference approach aims to speed up inference with less processing by integrating the inputs at later layers of a transformer decoder. An interesting research problem that we study in this paper is the impact of the LMM structure on the compression rate at a given distortion for distributed inference.



(a) Information-theoretic (best performance) approach



(b) Fast-inference approach

Figure 1: Different combining methods of the source and side information at the transformer decoder for LMM coding.

To take a step in this exciting research direction, we consider a single transformer encoder and decoder, each having its unique input type, as shown in Figure 1. The main problem is to accomplish the downstream task by filtering irrelevant features while preserving the relevant semantics to minimize the overhead. Within the context of LMMs, in Figure 1, the first input R can be regarded as an image of the transformer encoder and run on device A, which transmits its output to device B. The second input S can be a text prompt for the decoder and combined with the encoder output (e.g., a sequence of image tokens) on device B. Note that R and S are correlated sources with the alphabet $\mathcal{R}$ and $\mathcal{S}$, and a joint pmf $p(r, s)$ over $\mathcal{R} \times \mathcal{S}$, and generate a stationary and ergodic random process $\{(R_i, S_i)\}$ with $(R_i, S_i) \sim p(r_i, s_i)$. The alphabets $\mathcal{R}$ and $\mathcal{S}$ refer to the vocabulary files of the associated transformer encoder and decoder, respectively.

3 Semantic Compression Of LMMs

We propose a low-complexity algorithm based on randomly but hierarchically dropping some elements of the encoder output relying on the side information at the decoder. The dropped elements are indicated by the indices to the decoder to control the decoding complexity at the expense of some additional overhead. The hierarchy is utilized to minimize the overhead. Specifically, a hierarchical algorithm, named "*prune if not remove (PINR) by indexing*", is proposed to reduce the overall communication overhead of the encoder output in the presence of side information.

The main idea behind the proposed algorithm lies in first removing some of the tokens completely and then pruning some of the symbols of the unremoved tokens. The details are given in Algorithm 1. Specifically, in the first step, $\%f$ tokens are randomly selected and dropped. To facilitate the decoding process, e.g., avoiding iterative or any other complex operations, the index of each removed token is transmitted. Although this creates additional overhead, it is shown that its impact is negligible. In the second step, $\%g$ of the remaining tokens are chosen for pruning, and $\%h$ symbols of each selected token for pruning are dropped. Again, for decoding simplicity, the index for the pruned token and the corresponding indices of the dropped symbols are also transmitted. This hierarchical policy considerably decreases the overhead of the index transmission.

Algorithm 1 Prune if not remove (PINR) by indexing

Require: Denote a token by T
Require: Set the number of tokens to N_t
Require: Set the number of symbols per token to N_s such that $n = N_t N_s$
Require: $0 \leq f, g, h \leq 1$
Ensure:
if perfect reconstruction in the inner problem **then**
    $\mathrm{input} = \mathrm{X^n}$, $\mathrm{output} = \mathrm{X^{\tilde{n}}}$
else if imperfect reconstruction in the inner problem **then**
    $\mathrm{input} = \mathrm{X^n}$, $\mathrm{output} = \hat{\mathrm{X}}^{\tilde{\mathrm{n}}}$
end if
First (remove) step: token granularity
Randomly select $\%f$ tokens and constitute the set A such that $|A| = fN_t$
The set $I_A = \{I_1, ..., I_{fN_t}\}$ shows the indices of the tokens of $A = \{T_{I_1}, ..., T_{I_{fN_t}}\}$, where $I_k \in \{1, ..., N_t\}$ for $k = \{1, ..., fN_t\}$.
Remove the entire token in the set I_A.
Transmit I_A to the other device over the air, yielding an overhead of $|A| \log_2(N_t)$ bits
The remaining (unselected) tokens constitute the set B such that $|B| = (1-f)N_t$.
The set $I_B = \{\tilde{I}_1, ..., \tilde{I}_{(1-f)N_t}\}$ shows the indices of the tokens of $B = \{T_{\tilde{I}_1}, ..., T_{\tilde{I}_{(1-f)N_t}}\}$, where $\tilde{I}_{\tilde{k}} \in \{1, ..., N_t\}$ for $\tilde{k} = \{1, ..., (1-f)N_t\}$.
Second (prune) step: symbol granularity
Randomly select $\%g$ tokens in the set B and constitute the pruning set B_p such that $|B_p| = g(1-f)N_t$
The set $I_{B_p} = \{\hat{I}_1, ..., \hat{I}_{g(1-f)N_t}\}$ shows the indices of the pruned tokens of $B_p = \{T_{\hat{I}_1}, ..., T_{\hat{I}_{g(1-f)N_t}}\}$, where $\hat{I}_{\hat{k}} \in \{1, ..., N_t\}$ for $\hat{k} = \{1, ..., g(1-f)N_t\}$.
Transmit I_{B_p} to the other device over the air, yielding an overhead of $|B_p| \log_2(N_t)$ bits
for $\mathrm{T} \in B_p$ **do**
    Randomly prune $\%h$ of N_s symbols per token
    The set $I_s^{(T)} = \{I_{s_1}^{(T)}, ..., I_{s_{hN_s}}^{(T)}\}$ shows the indices of the pruned symbols per token T.
    Transmit $I_s^{(T)}$ to the other device over the air, yielding an overhead of $hN_s \log_2(N_s)$ bits
end for

The basic principle of the decoding algorithm is to acquire the (original or reconstructed) input source with the zeroized elements. This is obtained in two steps as follows. First, place the zeros to the corresponding locations according to the transmitted indices I_A, I_{B_p}, and $I_s^{(T)}$ for a given n. Then, the received tensor is placed to the remaining entities within the n-dimensional vector in order. After taking the encoded signal, it is combined with the side information either with the information-theoretic or fast-inference approach, and the transformer decoder runs by following the regular prefill and decode stages.

The communication overhead reduction comes from removing $\%f$ tokens and pruning $\%h$ of the total symbols of $\%g$ of the selected unremoved tokens. Let N_t denote the number of tokens and N_s be the number of elements or symbols per token. Then, the token removal saves fN_tN_s symbols, and the pruning results in a saving of $hg(1-f)N_tN_s$ symbols, where $0 \leq f, g, h \leq 1$. Considering N_b bits are utilized per symbol, the reduction becomes $(f + hg(1-f))N_tN_sN_b$ bits. On the other hand, the transmission of the indices leads to an extra $fN_t \log_2(N_t)$ bits due to the removed tokens, $g(1-f)N_t \log_2(N_t)$ bits to indicate the pruned tokens, and $hN_s \log_2(N_s)$ bits due to the pruning of symbols for each token, leading to $g(1-f)N_thN_s \log_2(N_s)$. This yields $fN_t \log_2(N_t) + g(1-f)N_t \log_2(N_t) + g(1-f)N_thN_s \log_2(N_s)$ bits. However, this is much less than the savings. For the sake of convenience, these calculations are simply visualized in Table 1, including the minimum required bandwidth[1] to serve the bits after the overhead saving under different SNRs. Since the total number of bits without comprehension is $N_t N_s N_b$, the reduction varies from $\%88$ to $\%22$ according to the selected example values of f, g, h when $N_t = 512$, $N_s = 1024$, and $N_b = 32$. It is apparent that any compression ratio can be synthesized by properly adjusting the values of f, g, h.

	Overhead w/o compression (bits)	**Overhead saving w/ compression (bits)**	**Ratio** (%)	**Min BW @0dB (MHz)**	**Min BW @20dB (MHz)**
{f, g, h} = 0.2	16,777,216	3.7229e+06	0.2219	13.05	1.96
{f, g, h} = 0.4	16,777,216	7.8152e+06	0.4658	8.96	1.34
{f, g, h} = 0.6	16,777,216	1.1723e+07	0.6988	5.05	0.75
{f, g, h} = 0.8	16,777,216	1.4894e+07	0.8877	1.88	0.28

Table 1: Overhead reduction ratios for some values of f, g, h when $N_t = 512$, $N_s = 1024$, and $N_b = 32$.

4 Results and Discussions

In this section, the performance of the proposed LMM compression algorithm is assessed over the Wyner-Ziv coding, both for the best performance and the fast-inference approach, for varying compression ratios. In our experiments, we use the LLaVa model of Liu et al. [2023] as the running example of LMM framework to validate the proposed compression algorithm without loss of any generality. The architecture of LLaVa relies on connecting a vision encoder, named CLIP studied by Radford et al. [2021], to a language model, named Vicuna in Chiang et al. [2023] via a projection matrix. In pattern recognition, relevance is typically quantified by precision and recall. The application of these metrics to generative models has been studied to measure the quality of the generated sample by Sajjadi et al. [2018], which formulates precision and recall to the relative probability densities of two distributions. In addition to precision and recall metrics, we use semantic similarity in our evaluations, quantified by the BERTScore, which correlates well with human judgment as stated in Zhang et al. [2020].

Figure 2 illustrates the performance of the compression algorithm under the transmission of a good channel (infinite SNR) for the two combining approaches, which are the best performance and fast-inference methods, in terms of precision, recall, and f1 score (or BERTScore). The results are normalized by taking the uncompressed signal as the reference. The important observations of Figure 2 are as follows. The first one is that even a slight compression of $\%30$ leads to a non-negligible loss. Secondly, and on the bright side, compressing further from $\%30$ to $\%90$ does not degrade the performance much. Combining these two results, we argue that it could be a good promise to sacrifice some performance with quite a significant compression, improving the communication overhead considerably, e.g., $\%90$. Finally, the best performance and the fast-inference combination show the same pattern, such that the latter provides computational complexity reduction at the expense of some additional performance loss. Specifically, one can argue that the fast-inference approach degrades the performance up to some point, depending on the compression. Notice that there is a negative bias toward the fast-inference method since the LLaVa model is trained according to the best performance case. That is, the multimodal architectures trained according to both cases at the beginning will result in more fair results, although there is no such publicly available model/checkpoint yet.

Next, the same experiment setup is repeated for bad channels and to emulate this scenario, the CLIP encoder output is transmitted at 0dB SNR (which is equivalent to a hypothetical low throughput channel according to the channel capacity formula). This case helps understand the robustness of LLaVa as well. To disentangle the impact of performance loss due to the channel conditions and compression, the compressed results are compared with 2 references: (i). the CLIP output is uncompressed and transmitted at 0dB SNR (labeled as “ref = 0dB SNR" in Figure 3); and (ii). the CLIP output is uncompressed and transmitted at infinite SNR (labeled as “ref = no noise" in Figure 3). More precisely, the former provides the real performance, and the latter is an auxiliary condition.

[1]Due to practical conditions, a larger bandwidth may be needed depending on many factors, e.g., modulation and coding scheme, signaling overhead, etc.

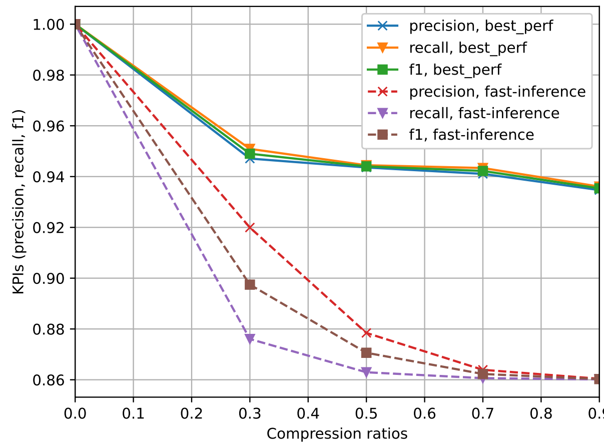


Figure 2: Transmission of the visual encoder output under compression for good channels.

As presented in Figure 3, in terms of the best performance combining technique, for (i), the poor channel conditions slightly decrease the KPIs for the wide range of compression ratios compared to Figure 2. Thus, the compressed source is robust to varying channel conditions. On the other hand, there is still nearly $\%8$ performance drop while going from no compression to $\%90$ compression. For (ii), the precision, recall, and f1 scores are mostly preserved under medium to high compression ratios with a loss of around $\%1$, i.e., compression brings almost negligible loss for bad channels. That is, the noise dominates the compression in the comparison of the noiseless-no compression case with the noisy-compressed case. Interestingly, as compared to the best performance case, the fast-inference approach becomes more robust to the poor quality channels.

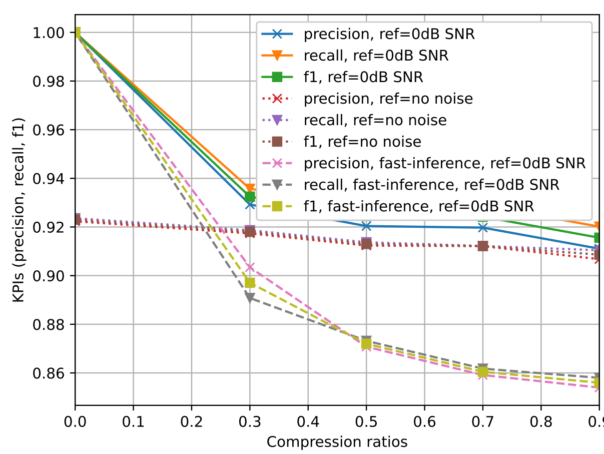


Figure 3: Transmission of the visual encoder output under compression for bad channels.

5 Conclusions

This paper provides a novel formulation of LMMs as a distributed coding framework by highlighting the connections to semantic communication. Specifically, LMMs that combine lossless access of text prompt with lossy compression of input-specific sources are a natural candidate for the constructive application of the Wyner-Ziv problem. Motivated by the large room for compression coming from the availability of side information, a compression algorithm is developed at the semantic level, and the results are extensively quantified with the recently emerged LLaVa model.

A Supplementary Results

To have a deeper understanding than the evaluated metrics, we also directly observe the generated text outputs of LLaVa since the best judgment can be made by the end users and the metrics may fluctuate depending on many factors, e.g., BERT architecture. To this end, we use the canonical example given in the original paper Liu et al. [2023]. Precisely, a nature view is provided with an input prompt to test image understanding, as given by Figure 4. Table 2 shows the response of LLaVa when the original CLIP output is transmitted without any compression and noise, and the $\%90$ compressed CLIP output is transmitted to the model without noise for the best performance information combining. Notice that the compression results in no semantic loss. The semantically relevant parts of the responses are preserved. This concludes that despite a quite aggressive compression ratio, the semantic information is kept, promising a significant savings in communication overhead. The same experiment is repeated with a different image in Figure 5, whose outcome is given by Table 3.

Figure 4: An example image view accompanied by a prompt to test image understanding.

Figure 5: An example plane accompanied by a prompt to describe the scene.

Without compression (original)	With compression
When visiting this location, which features a pier extending over a large body of water, there are a few things to be cautious about. First, be mindful of the weather conditions, as the pier may be affected by strong winds or storms, which could make it unsafe to walk on. Second, be aware of the water depth and any potential hazards, such as submerged rocks or debris, that could pose a risk to your safety. Additionally, be cautious of the tides and currents, as they can change rapidly and may pose a danger to swimmers or boaters. Finally, be respectful of the environment and other visitors, and follow any posted rules or guidelines to ensure a safe and enjoyable experience for everyone.	When visiting this location, which features a pier with a wooden dock extending out over a body of water, there are a few things to be cautious about. First, be mindful of the weather conditions, as the pier may be affected by strong winds, storms, or other adverse weather events. Second, be aware of the water depth and any potential hazards, such as submerged rocks or debris, that could pose a risk to your safety. Additionally, be cautious of the tides and currents, as they can change rapidly and may pose a danger to swimmers or boaters. Finally, be respectful of the environment and other visitors, maintaining a safe distance from the edge of the pier and avoiding any activities that could disturb the natural surroundings or other people.

Table 2: LLaVa responses for the original and compressed encoded image in Figure 4.

Without compression (original)	With compression
The image features a small white and black airplane flying over a city, with its landing gear down. The airplane is positioned in the middle of the scene, and its wings are visible as it soars above the urban landscape. The cityscape below consists of numerous buildings, showcasing the busy atmosphere of the area. The airplane's presence in the sky adds a sense of motion and excitement to the scene.	The image features a small propeller airplane flying low over a city, possibly during the day. The airplane is positioned in the middle of the scene, with its wings visible. The cityscape below includes buildings and streets, creating a contrast between the airplane and the urban environment. The airplane appears to be a small twin-engine plane, possibly a Cessna, as it flies over the city.

Table 3: LLaVa responses for the original and compressed encoded image in Figure 5.

References

W-L Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%

chatgpt quality. *lmsys.org/blog/2023-03-30-vicuna/*, March 2023.

G. Deletang, A. Ruoss, P.-A. Duquenne, E. Catt, T. Genewein, and et al. Language modeling is compression. *in Proc. Int. Conf. on Learning Representations (ICLR)*, 2024.

A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, and et.al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, Nov. 2024.

H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

E. Ozyilkan, J. Ballé, and E. Erkip. Neural distributed compressor discovers binning. *IEEE Journal on Selected Areas in Information Theory*, 5:246–260, April 2024.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, and et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, March 2021.

M. S. M. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly. Assessing generative models via precision and recall. *Conference on Neural Information Processing Systems (NeurIPS)*, Dec. 2018.

D. Slepian and J. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, 19(4):471–480, July 1973.

C. S. K. Valmeekam, K. Narayanan, D. Kalathil, J-F. Chamberland, and S. Shakkottai. Llmzip: Lossless text compression using large language models. *arXiv preprint arXiv:2306.04050*, June 2023.

J. Whang, A. Nagle, A. Acharya, H. Kim, and A. G. Dimakis. Neural distributed source coding. *IEEE Journal on Selected Areas in Information Theory*, 5:493–508, June 2024.

A. D. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, IT-22(1):1–10, Jan. 1976.

T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text generation with BERT. In *International Conference on Learning Representations*, 2020.