Audio-Journey: Visual+LLM-aided Audio Encodec Diffusion

Anonymous Author(s)
Affiliation
Address
email

Abstract

Despite recent progress, machine learning for the audio domain is limited by the availability of high-quality data. Visual information already presented in a video should complement the information in audio. In this paper, we leverage state-of-the-art (SOTA) Large Language Models (LLMs) to augment the existing weak labels of the audio dataset to enrich captions; we adopt SOTA video-captioning model to automatically generate video caption, and we again use LLMs to merge the audio-visual captions to form a rich dataset of large-scale. Using this dataset, we train a latent diffusion model on the Encodec embeddings. Furthermore, we leverage the trained diffusion model to generate even more audio data of the same format. In our experiment, we first verified that our Audio+Visual Caption is of high quality against baselines and ground truth (12.5% gain in semantic score against baselines). Moreover, we demonstrate that we could train a classifier from scratch using the diffusion-generated data, or use diffusion to enhance classification models on the AudioSet test set, working in conjunction with mixup or other augmentation methods for impressive performance gains. Our approach exemplifies a promising method for augmenting low-resource audio datasets. The samples, models, and implementation will be at `https://audiojourney.github.io`.

1 Introduction

The field of machine learning for the audio domain, despite making significant strides, is currently constrained by the scarcity of high-quality data. The largest datasets available, including AudioSet [8], CLAP [34], and VGGSound [4], comprise in total less than 3 million examples, falling orders of magnitude short of datasets in other domains, such as the Laion 5B Image-Text dataset [28]. Even for these datasets, the majority of annotations are collected through weak labeling, which may introduce noise. Our primary goal, therefore, is to significantly augment the limited audio resources.

To achieve this goal, we harness the power of generative models to create large amounts of diverse, high-quality audio data. Recent work has demonstrated the ability of Large Language Models (LLMs) to extract an enormous amount of knowledge from billions of text inputs [3, 32]. The in-context generation capabilities of these models are so convincing as to raise concern about their ability, "hallucinating" false responses that mislead human users [2]. As part of a careful data augmentation strategy, however, these hallucinations can be used effectively to generate captions that can significantly enrich the existing weak labels that annotate audio datasets.

The obvious concern is that such hallucinations could introduce even greater noise, especially when the original weak labels are vague. For example, a 10-second audio clip containing a multitude of sounds might be annotated with a single weak label of "speech." Replacing such a weak label with LLM output conditional on a single-word prompt might simply swap out one source of noise for another. However, datasets such as AudioSet are sourced from videos, with visual information

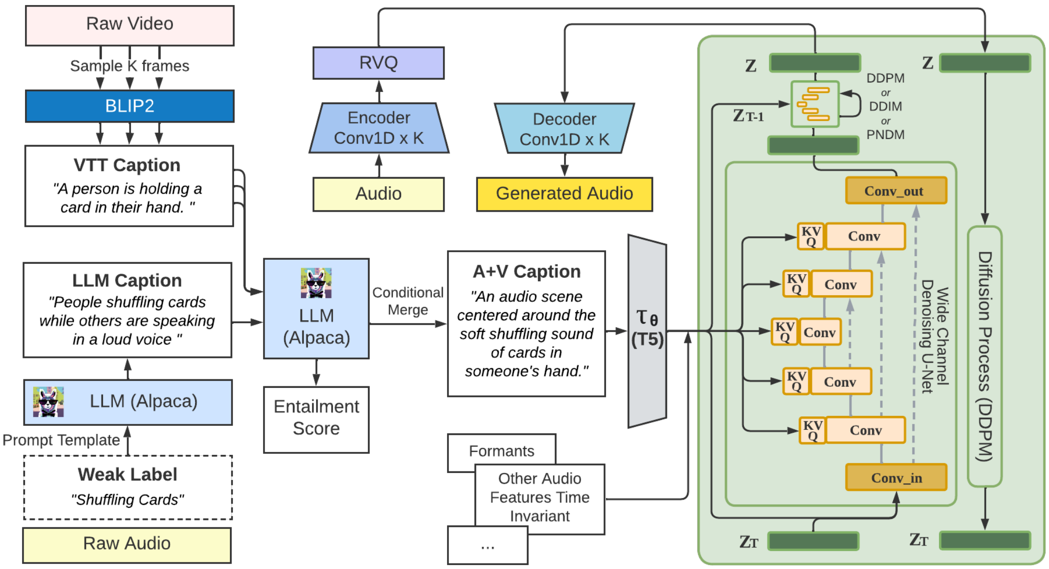

Figure 1: Our overall system diagram. BLIP2:[19], T5:[25], Conv_in and Conv_out layers are modified to 128 channels. Audio Encoder, decoder and residual vector quantized (RVQ) layers are pretrained by Encodec [6].

that can complement the audio. Video-to-Text (VTT) models can efficiently extract and transcribe visual cues into text representation, and state-of-the-art (SOTA) models such as BLIP2 [19] have demonstrated robust performance at this task. We apply BLIP2 to frames sampled from the video to generate a set of possible captions that capture visual information.

We have generated audio captions by prompting an LLM with the previously-available weak labels. Independently, we have also generated video captions using a VTT model. We once again leverage LLMs by computing entailment scores between these audio and visual captions to find the LLM "hallucinations" that are most compatible with the visual information. Additionally, we prompt the LLM to generate a merged audio-visual caption that provides an enriched annotation for the audio clip. Throughout this process, we remain particularly vigilant about potential audio-visual false-positives: some audio clips do not have any corresponding visual information, for example when the objects seen on video do not produce sound or when the sounds' sources are off-camera. We cover our approach to this issue in § 3. Our methodology so far is visualized in the left half of Figure 1, starting with the Raw Audio and Raw Video and leading up to the Audio+Visual (A+V) Caption.

Altogether, applying our methodology to AudioSet [8] produces a dataset of over 2 million audio clips with significantly-enriched captions. We perform a human evaluation to verify the quality of our generated captions. We find that our captions are quantitatively better than previously-generated captions [22] and qualitatively comparable to human annotations released by AudioCAPs [12]. Having constructed a substantially enriched audio-text dataset, we can learn a powerful generative model for audio. We encode each audio clip into a post-quantization embedding space [6], and train a score-based latent diffusion model to reconstruct the audio, conditional on a T5[25] encoding of our generated captions. We delve deeper into our modeling choices and motivations in Section§ 4. Our entire system is illustrated in Figure 1.

Our experimental results reveal that our diffusion model outperforms baseline models such as AudioLDM [20, 15] in generating higher quality outputs. Additionally, we prove that this model can replicate all the supplemental capabilities of Stable Diffusion [26], including features like ControlNet[37] and Dreambooth[27], among others[1].

To further validate the quality of both our generated captions and the trained diffusion model, we generate a large dataset of new audio samples and use it to train a classifier from scratch. Furthermore, we can also use our generated data to *supplement*, rather than replace, the existing AudioSet training

[1] Refer to the Appendix for more details of these add-on capabilities.

data; we show that the combination of real and generated data results in improved SOTA classification accuracy.

Our work, motivated by the need to augment low-resource audio datasets, makes the following contributions:

1. We release a new large-scale resource augmenting the existing AudioSet. This substantial increase not only enhances the volume of data available but also greatly diversifies it, providing a richer resource for future research and analysis.
2. We successfully train a diffusion model using a pretrained latent encoder-decoder, bypassing the need to train a VAE and vocoder (e.g., HiFiGAN [13]), which demonstrates excellent generation quality.
3. We showcase our diffusion model's ability to generate *useful* audio data. Classifiers trained with diffusion-generated data improve over those trained only on the original data.

2 Background & Related Works

Diffusion Models for Audio: Denoising Diffusion Probabilistic Models (Diffusion Models) [10] are a class of score-based generative models to predict how a data point diffuses over set time steps. The motivation for these models is as follows: given an image and a known forward diffusion process $q(\mathbf{x}_t|\mathbf{x}_{t-1})$, defined in equation (1), model and predict the reverse diffusion process $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, defined in equation (2). Once trained, the reverse diffusion process can map random noise into new samples from the training data's distribution. While accurately modeling the proper probability density function (PDF) of a sufficiently complex dataset $P(X)$ is intractable, diffusion models instead model the gradient or stein score of the PDF: $\nabla_x \text{log} P(X)$. Through integration, this score function conserves the information stored within the PDF without being intractable to compute [31], allowing for superior data coverage compared to other generative models. The forward process equation is as follows:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \qquad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{1}$$

where β_t is a value from 0-1 retrieved from the noise scheduler. Diffusion models have excelled at tasks including image synthesis [7] and audio generation ([20, 35, 15]). In contrast to other generative models, diffusion models suffer from a significant drawback: the extended duration required for sampling. This happens because the iterative denoising process requires multiple steps instead of a single forward pass employed by GANs and VAEs for generation. Many modern diffusion models address this limitation by operating in the latent space of an autoencoder, significantly reducing the dimensionality required for generation [26]. This approach improves image quality while simultaneously lowering sampling and training time. The reverse diffusion equation is as follows:

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \tag{2}$$

Where $\mathbf{x}$ is the noised latent, and t is the timestep, p_θ is an unconditional denoising model to approximate conditional probabilities, this model is parameterized through a score estimator $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$, which is equivalent to $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y = \varnothing)$. It's noteworthy that both p_θ and $\boldsymbol{\epsilon}_\theta$ can be effectively learned using a single neural network, in this work, we implement it with a high-channel UNet (more details in §4). Classifier-Free guidance is achieved by balancing the conditional and unconditional score estimator, where w is the tuning parameter, $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{T} \alpha_i$:

$$\begin{aligned} \bar{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t, y) &= \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - \sqrt{1-\bar{\alpha}_t}\; w\nabla_{\mathbf{x}_t} \log p(y|\mathbf{x}_t) \\ &= (w+1)\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - w\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \end{aligned} \tag{3}$$

Several recent works have used latent diffusion models for audio generation. AudioLDM and DiffSound [20, 35] generate audio by applying diffusion to spectrogram representations of sound. However, in addition to the denoising network, these approaches require training both a new VAE and an entirely separate vocoder (e.g., HiFi-GAN [13]) to convert from the generated spectrograms back into waveforms. This requires significant engineering effort and may be difficult to reproduce or generalize to new domains (e.g., if FFT parameters for spectrograms are hard-coded). In this work, we dramatically reduce the engineering effort and GPU hours needed to train an audio diffusion model. Rather than training our own VAE and vocoder, we use Encodec [6], an off-the-shelf VQ-GAN model

Classes	LLM Generated Captions
‘Singing’, ‘Yodeling’, ‘Speech’	‘A person singing and yodeling while talking.’
‘Pump (liquid)’, ‘Water’	‘The sound of a pump dispensing liquid and running water.’
‘Dog’, ‘Growling’, ‘Animal’	‘A dog growling and making animal sounds.’
’Frog’	’A frog is croaking in a dark, musty swamp.’

Table 1: Examples of conversions between class list and free text captions made to resemble image captions generated with an engineered LLM prompt.

which has demonstrated competitive MUSHRA [29] in high-fidelity audio generation. This allows us to focus all our training resources on the denoising U-Net. While using the pretrained VQ-GAN prevents us from jointly learning the latent space and the diffusion model, our model is still able to adapt to the Encodec model’s latent space. Our use of the Encodec model is similar to that of AudioGen [16], except they instead train an auto-regressive model. AudioGen also does not have public training code, making it a blackbox model and difficult to replicate.

Other Representation Learning Models: Directly modeling the score-based measure of the PDF allows diffusion models to significantly improve the diversity of their generation compared to other methods. For instance, GANs[13] model the generative pipeline without expressly capturing the underlying data distribution; while this can be sufficient for high-quality generation, it cannot fully capture the diversity of the dataset. VAEs are better than GANs at capturing the diversity of the data distribution, but often generate lower-quality samples [31]. Diffusion models generate samples that are of comparable or superior performance to GANs [7] while simultaneously producing better probability coverage of the underlying distribution [31]. Diffusion models’ ability to generate diverse and high-quality samples make them uniquely ideal for our goals of audio data augmentation.

Data Augmentation Methods: The conventional method of data augmentation in the audio domain involves utilizing basic transformations such as specAug [23] and mixup [36] to create fresh audio by masking and mixing existing ones. However, these augmentations lack diversity in semantics. The primary benefit of using diffusion models for augmentation is that we can address the low-resource nature of audio data and generate diverse samples of similar quality to the original data, given the diffusion model’s generation is high-quality and class-relevant. Recent work applying similar methods to the image domain suggests this is a scalable and accessible approach to data augmentation [33].

Automatic Caption Generation: Another common issue for audio datasets is the lack of high quality captions. Other efforts, such as WavCaps [22] and AudioCaps [12], have taken various approaches to this challenge. AudioCaps employed human judges to create audio-text pairs for over 46 thousand samples taken from AudioSet, whereas WavCaps used ChatGPT to generate captions based on the weak labels resulting in a new dataset of approximately 400k samples. Both methods fail to scale effectively due to the often prohibitive cost of human judges and premium closed-source APIs.

Previous Audio Classification: Previous works[18] on audio classification achieve high accuracy by modeling $P(y|X)$ with a typical supervised learning approach. Self-supervised (SSL) methods such as AudioMAE [11] incorporate a model of $P(X)$ through pretraining, but such a model is inappropriate for generation of new examples. Scored-based diffusion models model $P(X)$ directly and can use that understanding to directly perform classification. A properly trained diffusion model can function as a compressed knowledge base for the features provided in training [17].

3 Harnessing LLMs to generate Audio+Visual Captions: Prompt Engineering

Prompting given audio-only weak labels: We leverage the power of LLMs to increment the descriptiveness of the audio captions on datasets such as AudioSet[8], which only contains weak labels without descriptive captions. We use Alpaca [32] and engineered prompts to generate a richer caption for every sample in AudioSet balanced and unbalanced sets, unifying the list of audio classes and introducing the relevant concepts. Table 1 shows specific examples of the class list to caption transformation. Alpaca is an open-source instruction-following model fine-tuned on the Llama-7b model [32]. Utilizing this model generates more grammatical captions that remain faithful to the original labels.

To generate text captions from class label lists, we used the following prompt: *“For each of these, summarize the sounds into a single sentence: \n describe a situation with all of these sounds together:”*

followed by the clip's labels. A limitation of the Alpaca model is its tendency to add unnecessary details or ignore relevant labels when generating captions. By adding examples to the prompt, we leveraged the in-context learning ability of Alpaca to enrich our captions. Table A1 in the appendix covers more details.

Filter Hallucination and Obtain Visual Captions: Despite initial success with our approach, some captions contain Alpaca hallucinations, particularly in the cases of a single class caption. For example, in Table 1 line 4, *"swamp"* is a hallucination, however plausible. To address this, we filter captions to replace single-class captions with a simpler non-LLM derived caption *"The sound of [CLASS]"*. This second pass does not invalidate the Alpaca-generated captions, which are far superior at capturing the complexity of audio samples with multiple classes. Recognizing the lack of detail in this single class captions, we utilized a SOTA Video-to-text model BLIP2 [19] to generate video-based captions for each video. These captions were derived from 3 sampled frames within the video at the $\frac{1}{3}$, $\frac{1}{2}$, and $\frac{3}{4}$ points of the 10 second clips. We again used Alpaca to combine these three captions into one with the following merging prompt: *"Create one sentence that summarizes these three simply:"*, allowing us to more effectively summarize the information of each video from the frames sampled.

Merging Audio and Visual Captions: These video captions were then combined with the audio captions from single-class labels with Alpaca to provide the needed visual context within each caption and combat the hallucination generated with single-label Alpaca audio captions. In our prompt, we specifically focused on the audio-label while using the visual caption as auxiliary information: *"Summarize these two captions conditioned on the second caption, the second caption describes an audio class and is the main concept:"*. For all the prompts, we provided examples for Alpaca to better utilize the strength of in-context learning. Table A2, A3 in the appendix show more examples.

Audio-visual False Positive and Entailment Scoring: By observing the audio-visual captions generated while being aware of the audio-visual false-positive issue, we noticed that some of the captions in which the audio and video were not aligned resulted in noisy captions with details unrelated to audio in the actual clip. To measure the alignment between the visual caption and audio labels and test the capabilities of a decoder-only language model, we created an entailment score Alpaca prompt: *"on a scale from 0 to 1, output the probability that these two captions happen together in float format:"* In our in-context examples, we emphasized similar concepts even if the exact audio label was not referred to, such as an "ukelele" in audio and a "mandolin" in video receiving a high entailment score. Nonetheless, we maintain the integrity of the video caption by ensuring it did not introduce sounds not present in the audio label, such as the implication of speech through the depiction of a person, when the audio label did not include speech.

We sought to investigate the fidelity of our AV entailment score and overall usage of decoder-only LMs to calculate textual similarity and other metrics. In parallel, we use T5-sentence encoder [25] and the BERT sentence encoder and computed the embedding similarities between the audio and video captions. The Pearson correlation was then computed between the Alpaca entailment score and the scores from both T5 and BERT encoders to get 0.13 for T5 and 0.14 for BERT. We also calculated the (min, median, max) score for each metric with Alpaca having $(0, 0.55, 1)$, T5 having $(0.60, 0.75, 0.98)$, and BERT having $(-0.20, 0.34, 0.98)$.

These scores are not correlated which implies that a decoder LM may be less reliable than an encoder LM when determining textual similarity and calculating metrics. Despite this, using a caption similarity metric could ensure that future audio-visual training data is less noisy and has clearer relationships between different modalities and be used to guide the merging of audio-visual captions to determine when the visual context is useful to include in our caption.

Evaluation of Caption Generation: AudioSet's label coarseness and class imbalance are mitigated by our application of multiple Alpaca LLM caption generations, yielding 2.2M detailed audio clip captions. [2] Previous efforts on creating audio captions focused on automatic audio captioning[22], but with the small training set size and imbalanced classes within the dataset § 5, the performance of their model is also limited. These are also typically end-to-end with WavCaps[22] using a HTSAT-BART model [22] which lacks the explainability and scalability compared to our human language-focused approach with weak labeling. Our approach better considers different modalities and introduces more flexibility in label generation due to the controlled hallucination that Alpaca has.

[2] We will provide more details in the appendix and will release all captions as open-source resources.

Prompt Context	Similarity Score	BLEU$_1$	BLEU$_4$	METEOR	ROUGE$_l$	CIDEr	Vocabulary Size
Zero-Shot	0.474	0.128	0.006	0.079	0.128	0.094	-
One-Shot	0.605	0.178	0.014	0.100	0.182	0.193	-
Few-Shot	0.686	0.188	0.016	0.168	0.229	0.242	-
Audio-Visual Merge	**0.750**	0.165	0.013	0.109	0.178	0.183	**1480**
WavCaps[22]	0.667	**0.231**	**0.056**	**0.136**	**0.277**	**0.682**	509
AudioCaps[12]	N/A	N/A	N/A	N/A	N/A	N/A	871

Table 2: Average similarity scores ranging from 0.0 - 1.0 between captions generated with Alpaca prompts and ground truth captions (AudioCaps) where zero-shot means no examples given to Alpaca, one-shot is one example, and few-shot is several examples, automatic evaluation metrics compared to the ground truth, and vocabulary size for the sample captions generated.

To assess the performance of our captions and the improvements additional context provided in their generation we analyzed a subset of AudioCaps [12] captions (human generated captions) and their corresponding audio clips against our generated captions, scoring each on a scale from 0 to 1 on the similarity to AudioCaps[12] while referencing the actual audio clip as shown in Table 2. Since most automatic metrics are based on n-gram similarity or LCS [3] and generally do not perform as well on individual sentence comparisons [1], we decided to use a human metric because of the large vocabulary variation as shown in the subset vocabulary size shown in Table 2. Additionally, the WavCaps [22] model is fine-tuned on AudioCaps which helps boost their automatic metric scores in comparison to our approach.

These results clearly display the qualitative improvements gained from in-context prompting for LLMs with clear examples to guide text generation as well as the addition of visual elements into the captions. One limitation is the absence of labels in the original AudioSet which the human judges mentioned in the AudioCaps captions. Typically this occurred with speech and wind sounds being present in AudioCaps but not AudioSet labels. However, these cases show the benefits of using LLMs for generation which was able to use context clues and natural language understanding to add these missing features. An example of this context-aware enrichment can be seen in Figure 1 where the caption "shuffling cards" is correctly extended to include a human in the scene. Similarly, we observed that Alpaca would hallucinate "at the park" or similar setting-specific details for audio samples weakly labeled with ducks, water, or speech. Such hallucinations are often correct, but even when false they encode relevant domain-knowledge that helps improve the quality of our captions.

4 Text-guided Diffusion in Quantized Latent Space

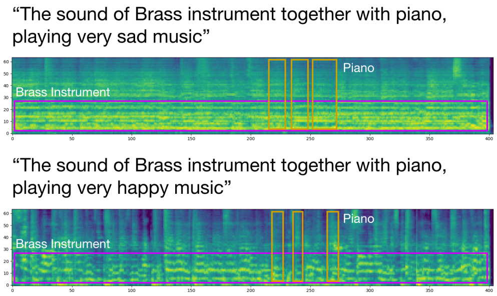

Figure 2: Spectrogram (only for visualization, never used) of sample generations showing tonal variance and harmonics.

Text Encoder τ_θ**:** We experimented with several text encoders for the prompt conditioned generation including CLIP [24], CLAP [34], and T5 [25]. The CLIP encoder we used to fine-tune our U-Net models complicated the domain shift from image to audio. While we initially used CLAP for its textual-audio joint embedding, we found it performed worse than T5. T5 has a larger embedding space than CLAP or CLIP, requiring an additional linear projection to connect it to the U-Net. We found this detail crucial to changing the text encoder while preserving pretraining knowledge. The second strength to T5's larger embedding space is its ability to encode larger vocabularies. As displayed in Table 2, our LLM-generated captions' vocabulary is significantly larger than that of either WavCaps or AudioCaps. To handle this more extensive caption vocabulary without loss, we chose T5 as our pipeline's text encoder. The final consideration for text encoding is using an attention mask on the text embedding. These encoders output a fixed-length embedding along with an attention mask usable to ignore invalid tokens. We experimented with and without attention masks with

[3] Longest common subsequence (LCS) and comparison of n-gram overlaps perform poorly with vastly different vocabularies and description styles as it is not able to capture semantic similarity well or match with completely different caption structures and scenarios

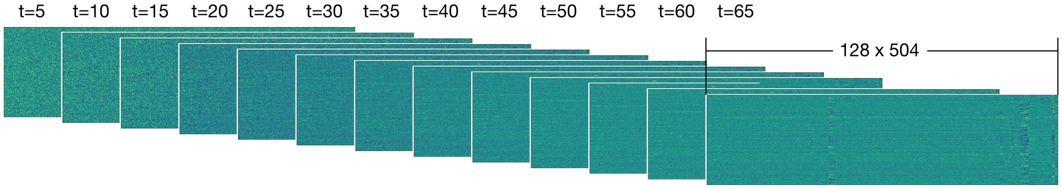

Figure 3: Latent space representations created throughout the denoising process. These images notably display the lack of interpretability in our generation space.

varying results. Experimentally, as shown in Table 4, masking had different effects on CLAP and T5-based models. Intuition would say that masking the T5 embedding would yield a more significant improvement as its fixed length is larger than CLAP; however, the addition of the linear projection layer between the text embedding and the U-Net functions as a type of masking resulting in inferior performance when combined with an attention mask, as was also discussed in [26] as "unmasked" expert model.

Noise schedule: The denoising process is trained across a fixed number T of DDPM [10] steps. A naive approach would be to simply compute noise percent as a linear (scaled) interpolation from 0 - 1 across the timesteps to control the β_t in Eq. 1, as done in AudioLDM [20]. With recent work [5] showing the superior performance of non-linear schedule vs. linear schedule, we adopt cosine noise scheduling without the need to tune the $\beta_{start}, \beta_{end}$ from $t = 0$ to $t = T$: $\beta_t = \text{Clip}(1 - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}, 0.999) \quad \bar{\alpha}_t = \frac{f(t)}{f(0)}$ where $f(t) = \cos\left(\frac{\frac{t}{T}+\delta}{1+\delta} \cdot \frac{\pi}{2}\right)^2$, where δ is a small offset. Figure 3 illustrates the process. During the sampling steps, we explored the usages of DDPM [10], and accelerated approximating schedulers: DDIM[30], and PNDM [21]. Our empirical results shows the best speed-quality tradeoff for PNDM [21], thus the results reported in the main paper are PNDM results. In the appendix, we include a detailed comparison of all noise schedulers.

U-Net ϵ_θ Design: We refer to our U-Net as a Wide Channel U-Net due to our choice to train and generate in a 128-channel latent space instead of the typical one or three channels used in SOTA audio generation. We had two main observations that informed this decision: first, the receptive field of the U-Net convolutional blocks could not fully explore the 128×504 latent space representations from the Encodec encoder; second, the latent encoding showed little variance within the 128 dimensions. We were able to leverage the second observation to correct the first by reshaping the latent vectors from a one-channel 128×512 image to a 128-channel 21×24 image. We then normalized each channel to a mean of zero and std of one representation to assist the U-Net in learning the noise: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. With this new representation, the convolutional blocks are able to contain the entire image in their receptive field without losing resolution and result in higher fidelity audio. After generation, these transformations can be fully inverted to allow for decoding back into a waveform.

Another difference between our diffusion approach and that of past work (e.g., AudioLDM [20]) is our use of cross-attention instead of embedding adding. This change enables us to conserve text embedding features. In self-attention, the text embedding is first concatenated to the image embedding, subjecting it to modifications at each layer of the U-Net. For cross-attention, we instead use the unmodified text for attention at each layer of the U-Net, maintaining the text embeddings fidelity throughout generation and improving class guidance. Equation 4 shows how this attention mechanism functions. In our cross-attention, the text conditioning y remains unchanged between diffusion steps; in self-attention, y would first be incorporated into the noised latent z_i before diffusion begins. In the self-attention setting, as the noisy latent passes through the U-Net the text embedding become increasingly interwoven into the noised latent and loses its reliability.[4]

$$\begin{aligned}&\text{Cross-Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right) \cdot \mathbf{V} \\ &\text{where } \mathbf{Q} = \mathbf{W}_Q^{(i)} \cdot \varphi_i(\mathbf{z}_i),\ \mathbf{K} = \mathbf{W}_K^{(i)} \cdot \tau_\theta(y),\ \mathbf{V} = \mathbf{W}_V^{(i)} \cdot \tau_\theta(y)\end{aligned} \tag{4}$$

Generation Latent Space: The Encodec model [6] we selected consists of an encoder, vector quantizer, and decoder stages. Initially, we attempted to directly learn the discrete "codebook" of RVQ as this has the highest degree of compression, at only 8×504, and could leverage the

[4] See the appendix for more detailed explanation, including specific dimensions, and potential pitfalls

generative benefits of the Encodec codebook and decoder stage. However, during experimentation, we observed nearly 0 decreases in train loss over time, as diffusion is only suitable for continuous vector space [31], an AutoRegressive model might be better for this. We trained our next model on the decoder embedding, which is of larger size, at 128×504, but is continuous. This slight change improved training substantially and resulted in the model contributions we are providing. Despite the discrete representation's inferior performance, we pre-computed the entirety of AudioSet 2M into discrete vectors and saved these new compressed versions to disk for training. This slight change from reading raw audio files to reading compact discrete codes from disk substantially accelerated training for two reasons. First, the I/O read times became significantly shorter as the files consist of 8×504 features instead of $160,000 \times 1$, resulting in a complete copy of AudioSet 2M that only takes 63 GB compared to 1.4 TB before for a $> 95\%$ reduction. Second, without needing to store these large waveforms in memory, we increased our batch sizes significantly, greatly improving training time. These results reinforce the vision of the original LDM in that it shows diffusion models can learn and generate arbitrary latent spaces regardless of their human interpretability or structure.

Diffusion As Data Augmentation: Using a pretrained text-to-audio diffusion model, we generated over 80,000 new audio samples randomly divided among the 527 audio classes in the AudioSet-20K balanced set. A random value $N \in [1, 2, 3]$ was selected and mapped to N random classes from the AudioSet-20K class list for each sample. With these samples generated, we combined them into the datasets shown in Table 3.

To closely match the original AS-20K dataset, these new samples do not have LLM-generated captions, and we generated them with a simple prompt formatted as "The sound of [LIST OF AUDIO CLASSES]." We chose this configuration to closely match the process used in [20] to convert labels to captions and provide prompts as close as possible to the captions on which [20] validated these models.

As shown in Table 3, we can see the classification models trained on augmented audio datasets improve with the growing size of the dataset. These results display the value of additional diffusion-generated samples as a form of data augmentation.

5 Experiment and Results

	Model	PT	Aug	AS-20kG	AS-20k	AS-40k	AS-60k	AS-80k	AS-100k	AS-2M
PANNs [14]	CNN	-	-	-	22.1	-	-	-	-	37.5
PANNs [14]	CNN	-	mx+sp	-	27.8	-	-	-	-	43.1
TALtrans [18]	CNN+T	-	-	-	22.4	-	-	-	-	38.3
TALtrans [18]	CNN+T	-	mx+sp	-	28.0	-	-	-	-	43.7
Our TALtrans	CNN+T	-	-	10.1±.50	22.4±.16	25.8±.11	27.0±.13	28.1±.06	30.1±.03	38.3±.15
Our TALtrans	CNN+T	-	mx+sp	11.2±.40	28.0±.20	29.4±.31	29.5±.13	30.7±.21	32.3±.14	43.7±.25
AST [9]	DeiT	-	-	-	14.8	-	-	-	-	36.6
AST [9]	DeiT	IM	mx+sp	-	34.7	-	-	-	-	45.9
Our AST	DeiT	-	-	3.2±.20	14.8±.17	15.4±.22	16.7±.14	18.1±.01	20.2±.18	34.6±.20
Our AST	DeiT	-	mx+sp	8.2±.61	16.9±.12	18.4±.32	19.5±.12	20.7±.22	22.4±.31	37.6±.10
Our AST	DeiT	IM	mx+sp	13.5±.50	34.7±.77	35.1±.18	36.1±.42	36.9±.03	37.5±.12	45.4±.70

Table 3: **Comparison with other state-of-the-art models** on audio and speech classification tasks. All datasets, other than AS-20k and AS-2M, are based from AS-20k with diffusion augmentation to add samples, details in section § 4. Metric is mean average precision (mAP). For pretraining (PT) dataset, AS:AudioSet, and IM:ImageNet. For augmentation (aug), mx+sp:mixup[36] and SpecAug[23]. Generation model and classification accuracy for each augmented dataset showing the improvements measured from diffusion as augmentation. Dataset AS-20kG consists exclusively of generated samples with 0 real samples from AudioSet. Our TALtrans Model (CNN+T: CNN+Transformer) has 12.1M params, and our AST model (DeiT/ViT-B) has 88M params.

Datasets: AudioSet contains 2 million 10-second YouTube clips, each weakly annotated for 527 types of audio events. Multiple events can occur in the same clip; a video of water boiling might be labeled with both "Liquid" and "Boiling." The data contains three splits: a class-balanced training subset (22K clips), an unbalanced training subset (2M clips), and an evaluation set (20k clips). The size disparity between training subsets highlights the underlying imbalance: there are over one million clips each labeled with "Music" or "Speech," but the rarest class ("Toothbrush") has only 127 clips.

AudioSet uses a hierarchical ontology[5] to categorize sounds; for example, "Toothbrush" is fully categorized as ("Sounds of things" → "Domestic sounds, home sounds" → "Toothbrush"). Despite the complexity of this hierarchy, many clips have only a single label that fails to capture the full complexity of the video's context. For example, a video of a toothbrush might be labeled simply with "Toothbrush" while containing the sounds of running water or speech.

We downloaded around 1.97M unbalanced training, 20K balanced training, and 19K evaluation clips. Some samples have been deleted from YouTube and could not be downloaded. For the AS-2M experiments in Table 3, we use the union of unbalanced and balanced sets for pretraining and fine-tuning. For the AS-20K experiments, we use AS-2M for pretraining and the 20K balanced set for fine-tuning. We report the testing mAP on the 19K eval set, and the same recipe as [18].

The primary differentiating factor between our method and those of other SOTA audio generation papers is our limited dataset. We trained our model exclusively on AudioSet with out Alpaca-generated captions. These results show the impressive capabilities of generation inside the latent space of the Encodec [6] model and our Alpaca captions.

Classification Accuracy: Table 3 lists classifier performance when trained on our diffusion-augmented datasets showing a clear image that additional samples generated with diffusion measurably improve classifier accuracy. Due to the large number of parameters of AST, it struggles to train from scratch, diffusion augmentation visibly alleviated this training difficulty and complemented pretraining. The most significant improvement comes from augmenting AudioSet-20K with an extra 20K generated samples, and the benefits slowly attenuate with more examples. However, it is important to note that diffusion cannot entirely replace ground truth data, as demonstrated by the inferior scores for AS-20kG. Nonetheless, it does yield measurable improvements when used as augmentation, especially when used in conjunction with other augmentation methods.

Generation Quality:

Model	**Datasets**	**Text**	**Params**	**FD**	**IS**	**KL**
DiffSound [35]	AS+AC	✓	400M	47.68	4.01	7.76
AudioGen [16]	AS+AC+8	✓	285M	-	-	2.09
AudioLDM-L-Full [20]	AS+AC+2	×	739M	23.32	8.13	1.59
Audio Journey-CLAP	AS	✓	861M	67.6	$1.63_{\pm.02}$	**0.127**
Audio Journey-CLAP-masked	AS	✓	861M	55.5	$1.64_{\pm.02}$	0.134
Audio Journey-T5-masked	AS	✓	861M	13.14	$1.64_{\pm.03}$	0.209
Audio Journey-T5	AS	✓	861M	**12.09**	$\mathbf{1.64}_{\pm.03}$	0.259

Table 4: comparison between our models and current SOTA models. These models are scored on frechet distance (FD), inception score (IS), and kullback–leibler divergence (KL). Our scores are computed by comparison against AudioCaps [12] test set explained further in § 5. Scores for DiffSound, AudioGen, and AudioLDM are from [20].

Table 4 presents interesting quantitative comparisons between our models and previous SOTA models. We performed our evaluation similarly to AudioLDM [20]; first, we extracted all captions from the AudioCaps [12] test set and generated samples based on each of these captions. We then compare FD scores against the ground truth audio from the AudioCaps [12] test set for each model, IS and KL scores are similarly measured. This shows two noteworthy trends: first, our generative model holds up to current SOTA models despite training exclusively on AudioSet with Alpaca-generated captions, whereas previous SOTA works include multiple other datasets. Second, the inverse trends of our FD vs. KL scores imply a trade-off between quality and diversity. This intuition is reflected in the models with said scores. CLAP model's superior KL scores are a reflection of the similarity between CLAP and CLIP, which these models were pretrained on. T5-based model's superior FD scores imply T5 assists in generation more than CLAP despite lower variance.

References

[1] George Awad, Keith Curtis, Asad A. Butt, Jonathan Fiscus, Afzal Godil, Yooyoung Lee, Andrew Delgado, Jesse Zhang, Eliot Godard, Baptiste Chocot, Lukas Diduch, Jeffrey Liu,

[5] https://research.google.com/audioset/ontology/index.html

Yvette Graham, , and Georges Quénot. An overview on the evaluated video retrieval tasks at trecvid 2022. In *Proceedings of TRECVID 2022*. NIST, USA, 2022.

[2] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623, 2021.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[4] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 721–725. IEEE, 2020.

[5] Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023.

[6] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

[7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[8] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE, 2017.

[9] Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*, 2021.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[11] Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. Masked autoencoders that listen. *Advances in Neural Information Processing Systems*, 35:28708–28720, 2022.

[12] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In *NAACL-HLT*, 2019.

[13] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33:17022–17033, 2020.

[14] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE ACM Trans. Audio Speech Lang. Process.*, 28:2880–2894, 2020.

[15] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis, 2021.

[16] Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. Audiogen: Textually guided audio generation. *arXiv preprint arXiv:2209.15352*, 2022.

[17] Alexander C. Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier, 2023.

[18] Juncheng B Li, Shuhui Qu, Florian Metze, et al. Audiotagging done right: 2nd comparison of deep learning methods for environmental sound classification. *arXiv preprint arXiv:2203.13448*, 2022.

[19] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[20] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D. Plumbley. Audioldm: Text-to-audio generation with latent diffusion models, 2023.

[21] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022.

[22] Xinhao Mei, Chutong Meng, Haohe Liu, Qiuqiang Kong, Tom Ko, Chengqi Zhao, Mark D Plumbley, Yuexian Zou, and Wenwu Wang. Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research. *arXiv preprint arXiv:2303.17395*, 2023.

[23] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin Dogus Cubuk, and Quoc V. Le. Specaugment: A simple augmentation method for automatic speech recognition. In *INTERSPEECH*, 2019.

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[25] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[27] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arXiv:2208.12242*, 2022.

[28] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

[29] B Series. Method for the subjective assessment of intermediate quality level of audio systems. *International Telecommunication Union Radiocommunication Assembly*, 2014.

[30] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[31] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020.

[32] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[33] Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. *arXiv preprint arXiv:2302.07944*, 2023.

[34] Yusong Wu*, Ke Chen*, Tianyu Zhang*, Yuchen Hui*, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023.

[35] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[36] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[37] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023.

A Appendix

Please see a separate document in the submitted supplementary material which includes the Broader Impact and Limitations sections.