Survey on Multi-modal Generative Models

Hans Brouwer HANS@BROUWER.WORK
Delft University of Technology, Mekelweg 5, 2628CD Delft, The Netherlands

Abstract

Multi-modal data synthesis is a burgeoning field as deep learning models have become able to synthesize high-definition samples in many different domains. These methods have great promise from both a creative and practical perspective in allowing creation of new data conditioned on already existing data. This survey seeks to understand general trends that are common across different approaches to multi-modal synthesis. We divide the problem into two parts: encoding/decoding multiple modalities and exchanging information across modalities. We find there are two prevailing approaches towards dealing with these aspects of a multi-modal synthesis task. Either a method makes use of modality-specific encoders and fuses the information from each modality using hand-designed processes, or a method employs a general purpose architecture which can learn how to process the different modalities on its own. Furthermore, this survey investigates some methods for self-supervision and unpaired learning in the context of multi-modal synthesis. These methods can help alleviate data-gathering requirements or strengthen training procedures to extract useful information with less explicit supervision.

1. Introduction

As generative models have continued to improve in past years, multi-modal models are coming into the spotlight due to their direct applicability to many varying tasks. Both multi-modal and generative models are overloaded terms. In this case, we mean multi-modal to mean multiple data modalities (e.g. text, audio, images, etc.) and generative to refer to models that synthesize new data. These models allow one to take one or multiple data inputs of different modalities and transform these to a new accompanying output.

Generally, multi-modal generative architectures can be separated into three main components: an encoder that translates data to a more useful condensed representation, an information exchange module which integrates the information from the different representations, and a decoder which translates the final representation to the output modality.

This survey will investigate these three components individually to compare and understand the different approaches for each. We will focus on modalities that have a time component (e.g. audio or video) and so will be targeted towards sequence to sequence approaches. However, insight from broader multi-modal literature (e.g. non-generative, non-sequence-based) will be integrated within the comparison of each component to understand what other possibilities are available.

To make this concrete, an example multi-modal task could be talking animal head generation. In this ex-

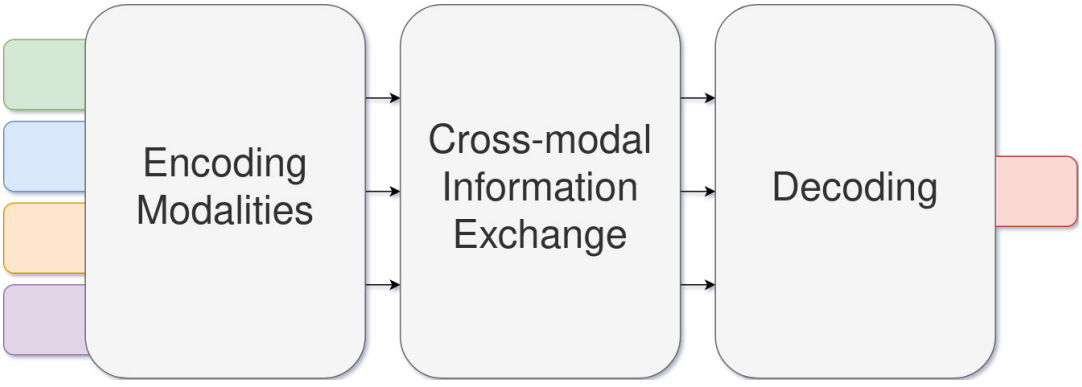


Figure 1. Generally, multi-modal generative architectures follow three steps: (1) encoding input modalities to some representation, (2) exchanging information across modalities, and (3) decoding to the desired modality.

ample text would be taken as an input and then videos of animals speaking would be the output. There are many possible architectures that could be used for this task, but generally the steps would consist of: (1) encoding the raw text to some vector format, (2) converting these text vectors to embeddings that integrate audio and video information, and (3) decoding the audio-visual representations to a final output video.

One interesting aspect of this task is that there is likely no good dataset of text transcriptions of talking animal videos. It is fairly common that gathering a large paired dataset is impractical, especially when considering multiple different data modalities. In these cases unsupervised or unpaired approaches might still allow a model to learn the desired synthesis task. Therefore this survey further investigates approaches which allow for unsupervised learning–especially for unpaired datasets.

First, we will dive into encoding and decoding of different modalities in the context of multi-modal networks. Oftentimes, encoding and decoding mirror each other closely and so can best be discussed together. Afterwards we will take a look at cross-modal exchange to understand how information from different modalities can be combined to synthesize new data. Finally, we will discuss the overarching themes and how the different pieces can be combined to create new powerful multi-modal generative architectures.

2. Encoding & decoding

The first step towards integrating information from multiple disparate modalities is to encode each to a format which makes them easy to compare. Frequently, input data is sparse in information and of an intractable size to learn from (e.g. high-resolution video). In these cases the data can be compressed to only the relevant information for the down-stream multi-modal process.

This generally requires domain-specific knowledge and specialized encoders for each modality. Many architectures simply leverage a pre-trained classification or feature-extraction network to condense the information [1][2][3]. However, recently a few modality-generic attention-based approaches have been introduced [4][5].

Similar to encoding modalities, decoding often must integrate domain-specific knowledge of the output modality. The architecture needs to generate the raw data, or some representation which can directly be converted to the raw data. Unlike encoders, this part often cannot be pre-trained (although we will discuss some exceptions) and so is the heaviest part of the generative architecture–in terms of both compute and parameter count. Once again, there are modality-generic decoding approaches which are analogous to the encoding versions [6][7].

2.1. Modality-specific en/decoding

The most straightforward pattern for encoding data is to use domain-specific networks for each individual modality. This allows for special handling of each modality so that specific inductive biases can be added where applicable. Furthermore, domain-specific encoders can be initialized from networks which are pre-trained on similar tasks, which can greatly increase training stability and reduce convergence time.

There are a few examples of these kinds of encoders in the multi-modal hashing literature. Multi-modal hashing tries to encode multiple different data types into a shared space for comparison (e.g. such that the word 'cat' and a picture of a cat have the same hash). While these aproaches are not directly applicable to multi-modal synthesis, encoding multiple modalities and finding a unified representation are important shared subtasks.

Li et al. [1], Xie et al. [3], Li et al. [2] and Gu et al. [8] all use variations on the same encoding scheme. CNN-F [9] pre-trained on ImageNet [10] to encode images and a bag-of-words approach to encode words to vectors. Li et al. [1] and Xie et al. [3] use one-dimensional CNNs over the word vectors, Li et al. [2] use a fully-connected network, and Gu et al. [8] a GRU [11]. Each then apply a different set of feature learning and/or adversarial losses between their encoded results to bootstrap a common embedding space (which will be further discussed in section 3).

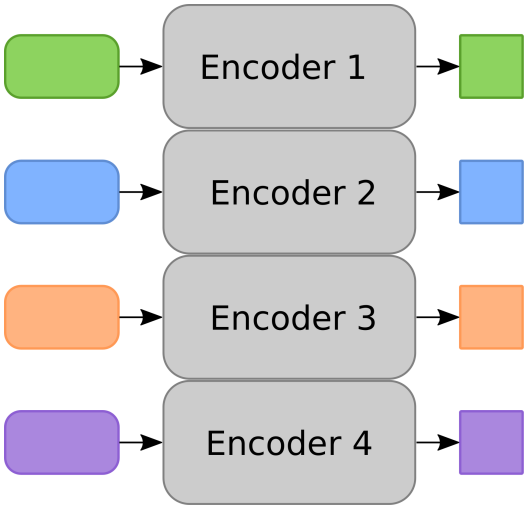


Figure 2. Modality-specific encoding schemes use separate, specialized encoders to compresss the representations of different data modalities.

Based on this shared encoding approach, they all share similar advantages and disadvantages. They all inherent powerful image recognition features to kick off their multi-modal understanding by leveraging a network pre-trained on ImageNet. Interestingly, they do not choose to use a pre-trained network for the text analysis. This is likely due to the large variation in the word vectors extracted from different datasets. Here a custom solution per dataset seems to be prefered. However, perhaps more general, large language model semantic embeddings could be used to good effect here (e.g. BERT [12]).

Outside the cross-modal hashing field, multi-stream encoders are also used in a generative setting. Peng and Qi [7] and Li et al. [6] apply modality-specific encoders to train conditional image generatrive adversarial networks (GAN [13][14]). Here the goal is to synthesize an image which corresponds to some conditioning information (text or electroencephalograms respectively). In both cases, the networks are trained as image auto-encoders, first encoding images and then decoding with the generator. Both approaches use a separate encoder to process their respective conditioning information (which is the second input to the generator) and apply a loss to match intermediate features of the two encoders. Peng and Qi [7] also add a text decoder which allows the final GAN to produce either image or text from a given image or text input. Li et al. [6] rather use the encoder networks to predict a label and apply extra supervision with a pairwise loss. Their cross-modal GAN architecture utilizes an interesting property of the multi-stream approach: the encoder streams are completely decoupled. This means that at test time only an EEG needs to be supplied to generate an image. However, it also means that unpaired images can be used during training. This allows the use of a large corpus of images without EEG recordings to augment training of the image generator. Both approaches add an adversarial discriminator to ensure that the generator is able to learn the desired distribution well.

These generative models also illustrate another interesting property of multi-modal generative architectures: often the decoder can allow for extra supervision which is not available to purely discriminative architectures. Using such an autoencoding scheme offers an element of unsupervised learning.

The clear weakness of modality-specific encoders is that they must be hand-engineered for each given modality. Small implementation details or improvements to each individual model might have a large effect on the overal systems performance. Furthermore, the number of modalities and interactions between modalities is fixed. Once the encoders and decoders have been trained together it is not simple to add new modalities. While these two problems might be alleviated by readily-available pre-trained models or freezing parts of the overal architecture to graft on new parts, it still leaves something to be desired from new solutions.

2.2. Pre-trained decoding

So far, we have seen that pre-trained encoders are often employed. It seems as if it would be hard to do something similar with the decoders due to the difference between encoded features and the usual inputs of unconditional decoder models (e.g. many GANs expect latent inputs of normally distributed vectors). Perhaps it would be possible to learn a transformation from extracted features to a standard distribution via some reparameterization approach (like in a variational auto-encoder [15]). However, a few architectures have been introdced that take a different approach.

Tian et al. [16] and Fox et al. [17] use a pre-trained image generator to synthesize videos. The image generator is used as a visual prior and a sequence of latent vectors is generated that correspond to a semantically correct video when converted to pixels by the generator. These architectures separate the learning of visual content from learning the motion of the video–delegating the majority of visual content to a pre-trained network.

Tian et al. [16] use StyleGAN2 [18] as the visual prior. This is a GAN which learns a hierarchical decomposition of the style of an image. The architecture uses a disentangled latent space, $\mathcal{W}$-space, where stylistic features of the image vary independently from each other along directions in the 512-dimensional space. A pair of LSTMs [19] is used to synthesize sequences of these latent vectors. One LSTM is only responsible for encoding a latent vector from $\mathcal{W}$-space to the hidden space of the other LSTM. The other then recurrently samples a sequence of latent vectors that will form the basis of the video. To ensure the content of the video is preserved, each vector output from the second LSTM is projected onto a PCA basis of the latent space and added to the previous latent along with a noise vector, ϵ, to model diversity of possible motion. The end result, after decoding the latents with the pre-trained generator, is a coherent video.

The latent LSTMs are trained using a supervised adversarial approach where a video discriminator and image discriminator try to distinguish between real/fake videos and video frames respectively. Furthermore a contrastive loss is added that uses frames within a video as positive samples and frames from other videos as negative samples to encourage videos to preserve content throughout.

The authors noticed that the LSTM often learned to neglect the noise vector, ϵ, and so added a final auxilliary loss that maximized the mutual information between the output latent and ϵ. The drawback of this training framework is the heavy adversarial learning being performed on high resolution video. This means the training set must contain a large variety of videos and training requires a lot of resources.

Fox et al. [17] alleviate this by training their motion generator purely in the latent space. The so-called StyleVideoGAN makes use of pixel2style2pixel (pSp) [20] which is an extension of StyleGAN2 which adds an encoder network that translates from image to latent space. pSp also makes the observation that the representation capability is higher when the entire latent hierarchy of StyleGAN is used. That is to say, rather than using one latent vector mapped to all convolutional layers of StyleGAN, learn a collection of latent vectors: one for each layer. This latent space is called $\mathcal{W}^+$. To generate the training data for the latent motion generator, training videos are encoded to a sequence of latent vectors by the pSp encoder. Once this is done, training can be performed without the StyleGAN network–greatly reducing computational requirements.

StyleVideoGAN's motion generator is itself also a GAN with a sequence of 4 GRU-blocks as the generator. The GRU generates recurrently in a 32-dimensional latent space. The output of each step of the generator is decoded to the full $\mathcal{W}^+$ vectors with a learned affine transformation, T. The first timestep is initialized with a normally-distributed latent vector, i, which is fed through an MLP, H, to initialize the hidden states of the first 3 GRUs. The last GRU gets i directly as the initial hidden state. The input to the GRUs are a sequence of 32-dimensional normally-distributed latents, $s_{0:t}$ which model the stochastic nature of video sequences. At each step, the hidden state of the final GRU is decoded to a full latent with, T, and the hidden states are re-used to decode the next s vector. Once the whole sequence is generated in this recurrent fashion, the discriminator (a six-layer MLP) tries to distinguish between real and fake sequences of latent vectors.

Both of these StyleGAN-based decoders offer an innovative way of integrating a pre-trained visual prior into a generator. A clear extension, is to condition the motion generator of such an approach on a sequence of multimodal information–enabling cross-modal video synthesis. What makes this approach even more interesting is that the pre-trained generator can be swapped out with another generator trained on different data to apply the learned motion to another domain (given that the other generator has a related latent space [21]). Paired with the entirely latent-based motion generator training by Fox et al. [17], this enables low-resource, small dataset motion adaptation without needing explicit cross-domain paired training data.

To illustrate this idea, we turn to the talking animal head example from earlier. A dataset of talking animals is hard to come by, but videos of humans talking are everywhere. First StyleGAN2 could be trained simply on human faces extracted from these videos (or simply use NVIDIA's pre-trained network for FFHQ [22]), then the pSp encoder to map videos to latent sequences, and finally the motion generator to learn latent sequences that look like videos of people talking. To translate these learned representations to animals, only the StyleGAN generator would need to be swapped out with one trained on AFHQ [23] (a high quality dataset of animals faces). Concatenate word vectors to the sequence of motion latents, s, and the our model can even perform text to talking animal head synthesis!

2.3. Modality-agnostic en/decoding

So far, the encoding and decoding approaches we have seen have all had one shared weakness: being locked into one single modality. To apply StyleVideoGAN to instead generate audio, first a high quality audio prior with a disentangled latent space must be found. Adding new modalities requires ad hoc approaches and domain-specific tricks. A truly multi-modal architecture which can be trained to encode or decode any modality is very desirable. Recently Jaegle et al. [4] propose exactly such an architecture called PerceiverIO (based on their earlier Perceiver encoder [5], extended to also decode).

The architecture is centered around self-attention [24], an operation which learns to transform and pass on information based on the inputs which correlate most strongly with each other. This creates a learning dynamic where the model automatically chooses which combinations of input values are important (which parts

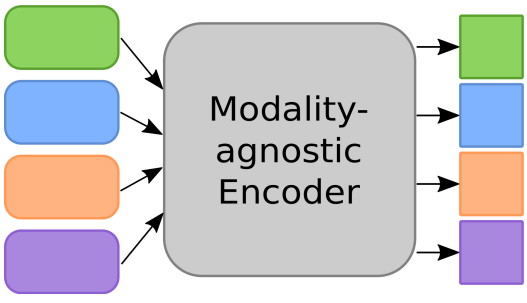


Figure 3. Modality-agnositc encoding schemes use a single unified encoder for all types of data.

it should pay attention to!). To be more specific, each attention operation consists of queries, Q, keys, K, and values, V. Generally these are all learned affine transformation of the input data. The output of the operation is the softmax of the correlation between the queries and keys times the values. V can be seen as information that should be passed deeper into the network with the interaction between Q and K determining which aspects are most important to pass forward.

Attention-based approaches have seen huge popularity since the publication of the Transformer architecture [24]–at first in the NLP community, but soon in other fields as well. One major drawback of self-attention is that it scales quadratically with input size. This means that such approaches do not scale well for large scale data like high resolution video or high sample rate audio.

The key insight of the Perceiver is to perform the attention operation between the input data and a fixed-size latent sequence to encode the data to a manageable latent dimension. This means that the computational and memory cost scale linearly with the input size rather than quadratically. The input to the encoding step is a number of sequences of vectors, M of some length, C, as well as N latent sequences of length D. Crucially, N and D can be chosen much smaller than their counterparts M and C. K and V are formed from the input data like usual, but Q is the latent array. This means the output of the Perceiver's attention operation is also an $N \times D$ latent sequence. This latent sequence is then processed by some number of latent transformer blocks using regular quadratic attention. The number of transformer blocks can be large because of the small size of the latent array.

In the original Perceiver architecture, the latent array is finally projected to some number of label outputs to perform classification. In PerceiverIO, Jaegle et al. [4] also introduce a method for decoding in a similar modality-agnostic way. The encoding attention step is simply reversed: K and V are formed from the latent array and Q from an "output query array" which has the same shape as the output data. Designing the output query correctly is still modality specific, however, significantly less research intensive than designing an entire modality-specific architecture. For example, the output query array for audio could simply be the timestamps of the samples or a 2D Fourier feature positional encoding [25]. For a text conditioned image to image application, the input array might be word vectors and the output query array could be the image to be transformed.

While finding the best way of constructing inputs and output queries might still require some trial and error, the architecture is incredibly flexible to encoding complex combinations of modality and task information. Rather than extensive feature engineering or preprocessing the data with separate modality-specific encoders, the raw bytes of each modality can just be concatenated and fed to the Perceiver directly. Another interesting aspect of this architecture is that the exchange of information between modalities is being learned completely by the attention operations. There are no hand-designed cross-modal inductive biases at all. While this seems like an advantage, perhaps finding ways to integrate some more biases into the Perceiver framework might perform even better. In the following section we will delve into some strategies for cross-modal information exchange that might be good candidates.

3. Cross-modal exchange

Exchange of information is the most crucial part of a multi-modal architecture. Designing an architecture such that it can effectively leverage multiple separate streams of information is the key to multi-modal synthesis. There are various approaches for cross-modal information exchange that are popular in different fields.

Once again we turn to the field of information retrieval where disparate modalities need to be compared. Cross-modal hashing accomplishes this by encoding data from various modalities to a shared hash space (usually a Hamming space) so that hashes can be compared for similarity. Approaches vary from multi-stage regressions that iteratively distill multi-modal information into the hash representations, to explicit separation of modality-specific and modality-agnostic data representations, to cycle-consistent generative adversarial networks.

Since the introduction of transformers [24], many approaches have investigated schemes for applying self-attention to distill cross-modal information into the learned representations. Transformers are sequence to sequence models, so much of this research focuses on how to merge information from sequences that stem from different modalities. Especially important is to do so efficiently as the quadratic scaling of self-attention with respect to sequence length can make naive approaches infeasible.

Contrastive learning approaches have also shown great promise recently. These approaches center around creating a shared space where similar examples are close together and dissimilar examples far apart. This approach is founded in mutual information theory and so can be leveraged even when modalities are not directly comparable.

Finally, we will cover approaches that allow for cross-modal exchange without explicit paired data. This can

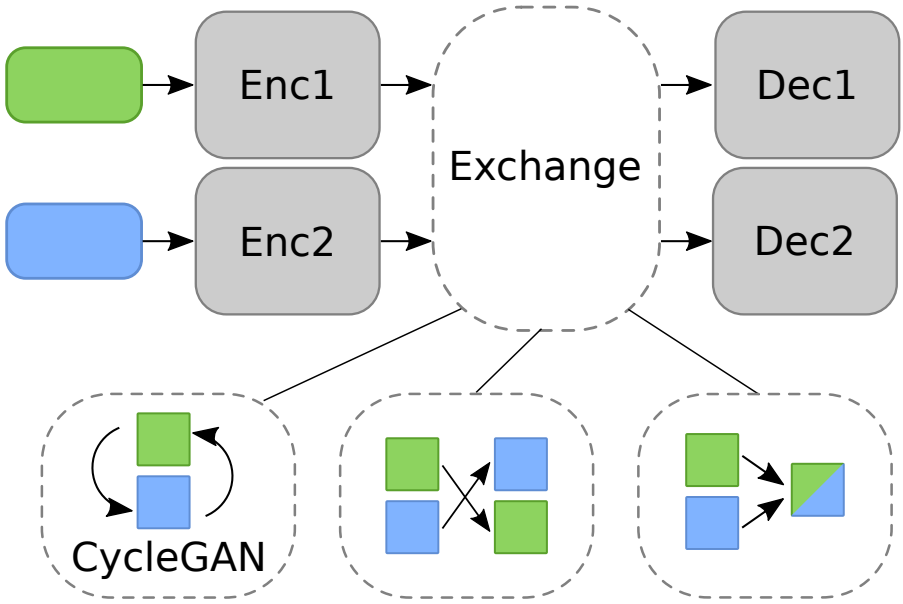


Figure 4. There are multiple ways to exchange information across modality streams.

often be valuable especially in cases where it is hard to find a large number of samples for a given modality.

3.1. Modality-common representations

Once again, we first look at methods for exchanging information between modalities condensed with domain-specific encoders. As we saw earlier, Li et al. [1] and Xie et al. [3] propose approaches which encode images with a pre-trained CNN-F network and based on 1D CNNs over bag-of-word vectors. However their strategies for merging these streams of information differ significantly.

Li et al. [1] use a self-supervised feature learning strategy. This "self-supervised" approach does in fact require labeled images and text for supervision, however not necessarily corresponding triplets. The key to the multi-modal representation learning is in the third encoder network trained in concert with the image and text encoders: the label network. The label network is an auto-encoder that reduces feature size to a bottleneck layer and then reconstructs the labels. The text and image networks are supervised with the intermediate features from the label network. An adversarial loss is added with discriminators that are trained in concert with the three encoders. The two discriminators learn to distinguish between features from the label network and from the image and text networks respectively. This further forces the network to create intermediate feature representations that were shared between the text and image encoders.

Xie et al. [3] also use such an adversarial approach to create a modality-common representation from the encoded features, however, do so without use of a third label encoding network. This approach is rather based around explicitly splitting out modality-common from modality-private representations. This is achieved through the use of three classifiers. The first, semantic consistency discriminator which tries to classify modality-private and modality-common representations as 'real' and 'fake' respectively (while the encoders are trained to make the discriminator classify as the opposite). This ensures that the modality-common and modality-private representations are similar. The second classifier learns to classify the representations as either image-private, text-private, or common. This tries to ensure the modality-private representations of text and images are different. The final classifier uses each of the representations to try and classify the corresponding labels of the original data instance. This is to preserve the semantic information in the shared representation.

While the two approaches take different strategies to distill the information from both modalities into a single unified representations, the underlying strategy is similar. They use adversarial classifiers across the modal representations to promote certain aspects in the final learned embeddings. In a generative setting, these same approaches can be applied to intermediate features to make sure that the decoders work from a similar unified understanding of the input modalities.

Peng and Qi [7] use a two stream adversarial auto-encoder for text and images. Each modality is first

encoded to a latent bottleneck representation, transformed to a common representation by a set of shared weights, and then finally decoded back to the original modality. The weight-sharing constraint on the feature transformation layers has a similar role to the feature-matching losses used by Li et al. [1]. Furthermore, an intra-modality discriminator is added to each modality's stream to try to distinguish between the encoded representation and the common representation after transformation, as well as a discriminator that tries to classify whether the common representation comes from the image stream or the text stream. This mirrors the adversarial classifiers used by Xie et al. [3]. Finally a reconstruction loss is added to ensure the images or text generations are faithful to the originals.

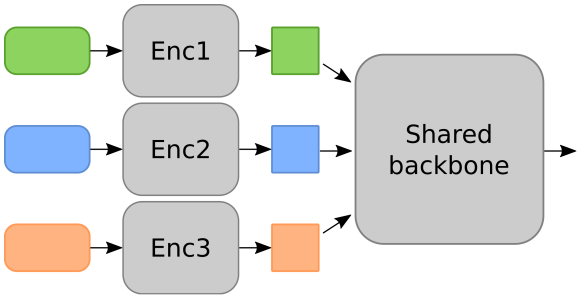
Li et al. [6] introduce a semi-supervised cross-modal GAN that generates images conditioned on electroencephalograms. Rather than using an auto-encoding approach for both streams, they supervise both encoders with a label classification loss. To achieve a shared semantic representation, they simply add a loss between a fixed layer in the image and EEG encoders. To generate the image, they concatenate the predicted label and the semantic feature layer. The image generator is supervised by an adversarial reconstruction loss as well as an image-quality adversarial loss (i.e. a discriminator which classifies a set of real and fake images).

The generative approaches of Peng and Qi [7] and Li et al. [6] highlight how ideas from tangentially related, non-generative approaches can be used to great effect in a generative setting. Exchanging information between separate modality streams into a common representation is an effective way to synthesize data conditional on multiple data modalities. Next we will take a look at some methods which seek to reduce the explicit inductive biases present in these hand-designed multi-modal architectures.

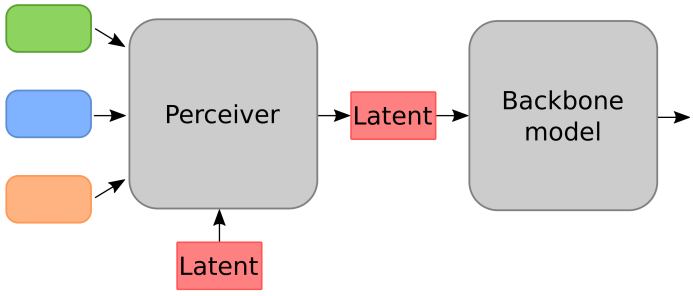
3.2. Attention-based approaches

Since the publishing of the seminal papers on Transformers [24], the architecture has permeated into all sorts of different applications in many different fields and modalities. As discussed earlier in subsection 2.3, the architecture relies on the self-attention operation to determine which aspects of the input data are useful for the task at hand (n.b. the transformer is simply an expressive architectural block built arround self-attention, so we will simply refer to the general concept as 'attention'). Rather than specifying explicit inductive biases (e.g. the correlations between nearby data elements are important, in the case of convolutions), attention lets the model learn the most important correlations on its own. This immediately seems promising in the context of multi-modal architectures as it would simplify the design of cross-modal information exchange to simply applying attention between modalities. Of course, the details of how attention is applied are quite important. Multiple groups have proposed different ways of using attention in a multi-modal context and investigated the tradeoffs of different strategies.

Radford et al. [26] introduce Contrastive Language-Image Pre-training (CLIP). CLIP uses two separate encoders for text and images to learn a shared embedding space where the similarity between text and images can be determined by taking the cosine distance between the respective embedding vectors (essentially a form of cross-modal hashing!). This is achieved through contrastive learning which will be discussed in subsection 3.3, for now we focus on the encoders. Radford et al. [26] experiment with a few different encoders for both text and images: bag of words or transformer-based for text, and ResNet-based [27] or vision-transformer-based (ViT) [28] for images. In the end, the transformer-based approaches are the strongest versions published in the official open-source implementation [29] of CLIP. This encoding approach is essentially equivalent to the multi-stream encoders we have seen already in that it fuses the



(a) Late fusion: modalities are first encoded with domain-specific encoders and then fed through a shared backbone.



(b) Early fusion: different modalities are immediately fused together and fed through one unified pipeline.

modalities late in the process.

Akbari et al. [30] try a different approach at integrating the information from the different modalities (although they apply a similar contrastive training method). They encode video, audio, and text (hence the name Video Audio Text Transformer or VATT) using domain-specific encoders, but then feed all modalities through a single shared transformer–an early fusion approach. The authors also investigate using separate transformers per modality (i.e. only merging in the final contrastive step) and find that the shared-transformer is on par with this approach. To tame the quadratic scaling on large-scale (video and audio) data, Akbari et al. [30] introduce DropToken. The idea is to randomly drop a percentage of the sequence inputs to the transformer, reducing the size of the sequence. They find that the trade-off between quality and performance is favorable and are able to drop half of the tokens without a significant effect on training.

Nagrani et al. [31] take a systematic approach to evaluate the effects of fusing multi-modal information early or late in the transformer. They do this in context of their proposed attention-bottleneck transformer which uses separate streams for each modality but with a cross-modal exchange stream. This is a smaller latent sequence that both streams perform cross-attention on. The modality-specific streams never explicitly attend to the other, but can only communicate through the bottleneck. This is much more efficient than full cross-attention between modalities. An ablation on latent dimension finds that performance with a size of 4 is as good as latent sizes up to 256. A second ablation finds that the best performance is found when starting cross-modal exchange in block 8 of 12 total transformer blocks. This lets the model create an effective sequence representation of each modality individually and then finally merge these together for final multi-modal sequence processing.

All of three of these methods take steps toward the goal of replacing cross-modal exchange with attention. Perhaps combined with Perceiver-based [5] encoding, they would come even closer to a fully modality-agnostic architecture by removing the need for DropToken or latent bottlenecks to tame quadratic scaling. However, for now, all three share one large drawback: requiring large-scale pre-training. The largest CLIP model took 18 days to train on 592 V100 GPUs [26] with a dataset of 400 million image-text pairs scraped from the internet manually and not released. VATT was trained with 256 TPUs (v3) in 3 days [30] on 136 million video-audio-text triplets scraped from the internet or sampled from AudioSet [32]. The enormous scale of compute and data required for the models to learn effectively puts these methods squarely out of reach for many applications. While some pre-trained CLIP models and inference code have been published, it is still difficult to adapt them to new domains as training code was not published. This highlights the need for further scaling down the size requirements of transformer-based approaches, however, for now we must make do with what we have.

Sollami and Jain [33] offer one interesting approach at repurposing a pre-trained model. They introduce MAnTiS, Multimodal Adaptation for Text Synthesis, a framework for incorporating multi-modal conditioning information into a pre-trained natural language generation model. The approach is to first encode each modality of conditioning through modality-specific encoders, then project these latent vectors to text tokens, and finally concatenate them together with the regular inputs to the language synthesis model. This allows arbitrary modalities to be used under the same scheme and does not introduce any extra parameters or interactions within the pre-trained generative network (which might degrade generation performance). The encoders and generative model can all be fine-tuned from domain-specific pre-trained networks, keeping the required computation low. A similar scheme could be devised using a multi-modal encoder like CLIP or VATT instead of domain-specific encoders.

Such fine-tuning approaches that can repurpose generic pre-trained models to accomplish new tasks are a promising way of democratizing the impressive performance benefits of large-scale models. Another nice benefit of such large models is improvements in self- or unsupervised training approaches. Oftentimes, these are also portable to smaller-scale problems. One such technique which has clear applications in multi-modal contexts is contrastive learning.

3.3. Contrastive learning

Contrastive learning [34] is a training scheme which supervises a model by showing it positive and negative examples. The key insight is that positive samples and negative samples can often be generated directly from a batch of unlabeled examples. For example, positive samples could be multiple augmented (flipped, contrast-enhanced, cropped, etc.) versions of the same image and the negative samples all other images in the batch. This allows for bootstrapping representation learning purely from unlabeled data and then using the condensed representations efficiently in downstream tasks.

Alternatively, in the presence of labels or pairs, positive and negative samples can be drawn according to the extra information. Radford et al. [26] and Akbari et al. [30] both use a scheme where pairs or triplets of data samples

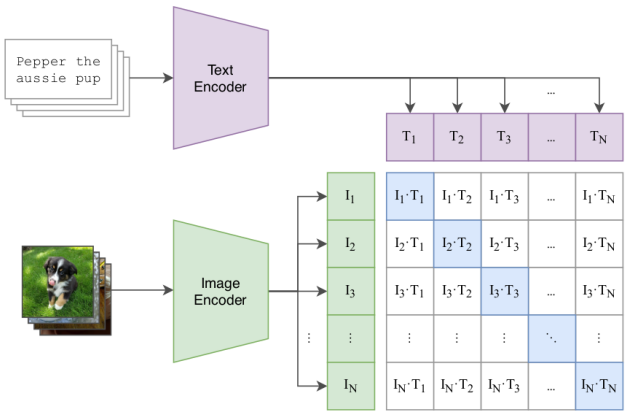


Figure 5. Contrastive training between text and images. Postive samples are shown in blue [26].

from different modalities are used to train an embedding that allows for comparing similarities between samples from any of the modalities. Both make use of a noise contrastive estimation loss [35] (specifically, InfoNCE [36] in CLIP and NCE + MIL-NCE [37] in VATT) which tries to classify positive samples together and negative samples apart. In CLIP's contrastive pre-training, the positive pairs are simply the text embedding corresponding to each image embedding. For VATT there are extra modality-specific projection heads that convert the output from the transformer to a specific modality embedding. Between video and audio features, a simple NCE loss is used (as the correlation between these two modalities is presumably high). To compare with text, the video embeddings are first projected to the audio-visual comparison space and then projected once more into a space where MIL-NCE is used to compare with the text embeddings. This hierarchical space help to bridge the difference in scale between audio-visual data (which is very large and relatively information-sparse) and text data (which is small but information-dense).

Lee et al. [38] take a different tack. Rather than using contrastive estimation to train a model, they instead use it to curate dataset by estimating audiovisual correspondence in videos. They apply multiple pre-trained models–image recognition (ResNet-50 [27]), video action recognition (SlowFast [39]), and audio classification networks (VGGish [40])–to extract intermediate network features of the videos. The authors apply PCA to reduce the feature representations to one-dimensional fixed-size embeddings to keep things computationally manageable. To find videos with high audiovisual correspondence, they train linear projections to transform each modality to a shared embedding space using contrastive learning. The positive samples are simply the video and its audio. Once trained, the cosine similarity between the projected audiovisual embeddings approximates the mutual information between the two signals. The best performance is achieved when approximating the mutual information between all combinations of the five feature layers per network (e.g. the shallowest layer of ResNet against all other layers in SlowFast and VGGish, the second shallowest against all other layers, etc.).

While the previous methods are not used in a generative setting, similar approaches can be useful. For example, Park et al. [41] use contrastive learning to learn unpaired image to image translation (e.g. translating pictures horses to zebras). They introduce a generator that consists of an encoder and a decoder with a multi-layer patchwise contrastive loss to maximize mutual information between the input and target domains. The approah takes L layers of interest from the generator's encoder and the final pixel output, then passes each through a 2-layer MLP projection head. Then spatial patches are taken from these partially-encoded projected features and used in a contrastive loss (InfoNCE). The positive pairs are the patches of the same region in the extracted features and the negative pairs are the rest. The idea is that there should be a high mutual information between the different representations of the same patch of an image throughout the network. This ensures that the output image has a similar structure to the original image. To ensure the image translation also learns to transform to the new domain, an adversarial loss is added that tries to distinguish translated images from real images in the target domain.

This approach shows the promising technique of discriminative or encoding models supervising the representations in the generative part of the pipeline. One issue here is that it is not immediately clear how this approach can support multi-modal data. Across some modalities it is immediately clear which patches could be positive samples (e.g. patches of time in audio and video), but for others it is not (e.g. text and image). Similarly, when contrastive methods rely on augmentations to form positive samples, there must be reasonable augmentations to apply. Designing effective augmentations is a topic of its own which can make it hard to apply contrastive learning to modalities where augmentations are not yet well-studied.

Towards investigating and alleviating these issues, Patrick et al. [42] create a framework for generalized data transformations (GDT). This framework generalizes a number of popular augmentations strategies under one unified format. While the framework does not introduce universal augmentations that can be applied to anything, it creates a guideline for how to design sets of augmen-

tations in a multi-modal context such that the correct invariances (transformations under which positive samples remain positive) and distinctions (transformations under which positive samples become negative) in the data are captured. A GDT consists of a sampling operation, a set of augmentations, and a contrast function. The contrast function defines whether two samples under a set of augmentations are positive or not. By designing the contrast function one can specify wide range of relations that the model should learn about the data. The authors give multiple examples showing how many contrastive learning approaches proposed in the literature can be respresented simply and compared using this formalism.

While GDTs do offer a better understanding of the structure of contrastive relations, they still rely on sets of hand-designed augmentations being available and being compatible between the specific set of modalities in a task. Verma et al. [43] propose an approach which attempts to achieve domain-agnostic contrastive learning. The basis of the technique is to use Mixup [44] to create positive samples: simply put, randomly interpolating two samples. The authors show that a contrastive approach centered around these interpolations is effective at learning representations. The crucial step towards being domain-agnostic is that the mixup interpolations can also be applied to deep features. This means the only constraint is that the two domains can be encoded to tensors of the same shape. They further give a variety of interpolation types (ranging from binary to uniform to geometric) and test them on multiple different problems in different modalities.

Such innovations provide a clear path to contrastive learning becoming a dominant paradigm in multi-modal tasks. Being able to decouple training procedure and losses from the specific modalities that are being processed yields similar benefits to making architectures domain-agnostic. The more generic and effective the model, the more useful it is in practice. One other frontier that can provide important benefits in practice is data. Many training techniques (including contrastive ones!) rely on explicitly paired information to extract a useful representation. Often this is hard or impossible to come by. Next we will discuss some techniques that allow for learning with unpaired, multi-modal data.

3.4. Unpaired data

As already touched on a couple times, being able to train with unpaired data makes an algorithm quite a bit more flexible–especially in a multi-modal context. However, learning useful representations from unpaired data is difficult without some clear connection between the different modalities. Often this connection can be inferrred through use of labels which are common to all modalities or though an extra modality that bridges the gap. On the other hand, there are a set of approaches centered around cycle consistency which learn to translate from unpaired data by converting data back and forth.

3.4.1. Label supervision

Most machine learning is done on sets of labeled data. Therefore it is often possible to find some dataset that captures an aspect of a given task. However, the more modalities that are added to the mix, the less ready-made datasets are avaible that cover all modalities. Luckily, if there is overlap in labeled concepts it is possible to bridge multiple separate datasets and bootstrap multi-modal learning.

Gao et al. [45] introduce a multi-step process for cross-modal hashing which has a closed form for each individual step. The target domains are images and text which have been labeled with concepts that are present in each. The approach divides the hashing problem into multiple matrix factorization and regression tasks. First, kernel logistic regression is used to perform collective matrix factorization on the two modalities. The goal is to find a common space, V, and two basis matrices, U_1 and U_2, for each modality. The common space V are used as initial hash codes for each sample of the modes and will be iteratively improved. Next, the similarity matrix between the k nearest neighbors (by L_2 distance of their vectors in V) is minimized. Then, a rotation matrix is regressed to minimize the error of quantizing to hashes. Finally, to include information from the labels, the label matrices and an afinity matrix (pairwise similarities between labels) are used as minimization targets of the quantized hashes. This multi-step process gives the final hash codes which can be used to compare data from the disparate modalities. To perform inference on new data, a final linear regression is trained on the input and output of the multi-step optimization process that can approximate the hash codes for unseen data.

In the days of deep learning, it is refreshing to see that well-designed, closed-form statistical learning holds its own on even on complex multi-modal tasks. Such an approach could be used as input features to a synthesis algorithm or some of the intermediate optimization targets tweaked to directly construct a desired output signal.

3.4.2. Skip-modal supervision

Another option, when labels are not available to bridge the gap, is to use an intermediate modality. Ma et al. [46] bootstrap cross-modal unpaired synthesis under the

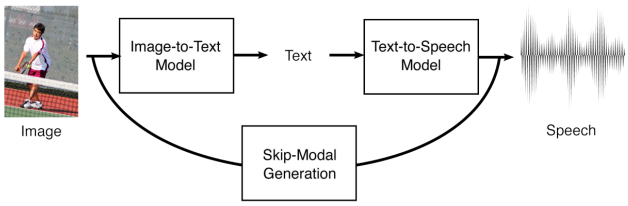


Figure 6. Learning image to speech synthesis without paired a image-speech dataset [46].

assumption that there exists paired data but not between the modalities that are desired. This weakens the requirement of labeled examples to, for example, examples being described in words. The authors specifically use image-text and text-speech datasets to learn a direct image-speech translation.

The approach is to train an auto-encoder with encoders and decoders for each modality with a shared attention-based multi-modal bottleneck. The initial encoded representations from each modality-specific encoder is further encoded to a modality-agnostic embedding. An adversarial classifier is used on the modality-agnostic representation to classify the modality (encouraging a uniform representation). The main multi-modal exchange is performed in the "memory fusion module" which extracts abstract concepts from the unified representation. This is achieved through multi-head attention across a learned memory bank. The modality-agnostic representation is the query, keys are extracted from the memory by a 1D convolution over the memory vectors, and values are the actual memory bank contents. This transforms each modality-agnostic embedding to a new representation which the decoders can use to synthesize the original data. A reconstruction loss is put on the outputs of the decoders versus the inputs.

3.4.3. Fine-tuning

In both label-based and skip-modal unpaired learning, there is still some amount of paired data between modalities. Another view of the unpaired task is to avoid having samples that are paired. Deng et al. [47] investigate such a task. Their goal is to synthesize talking heads based on speech input without any pairing between input speech and output video. They train an auto-encoder to synthesize both audio and video based on an audio input. Then, to synthesize video for a new speaker, they finetune the auto-encoder to a small dataset of the new speaker. The encoder and decoders can then be mix-and-matched to translate between speakers. The audio encoder is taken from the "input" speaker and the audio-visual decoder from the "output" speaker. This allows for talking head synthesis without any inter-speaker paired data.

[21] show that similar latent space re-use between pre-trained models can be applied in a StyleGAN architecture. They find that finding meaningful latent directions in the pre-trained model are transferred to fine-tuned children models. This means that finding paired data in one domain to analyze the latent space is enough to be able to apply the latent translations on a model fine-tuned to a new dataset. These findings are also mirrored by the video domain transfer in [17], which uses this same property of StyleGANs.

3.4.4. Cycle consistency

Next we look at a few approaches which learn to translate data between different domains and modalities with only unpaired data using cycle consistency.

This was originally introduced by the seminal paper "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks" [48]. Cycle consistency was originally applied in the image to image case. Two separate image-to-image GANs are trained: one which translated from image domain A to B and the other which translates back. Each sample is then transformed to the target domain, B, and back. The cycle consistency loss tries to ensure that the output image is unchanged after being transformed twice–ensuring that the two learned transformations are each others opposites. To prevent learning a simple identity function a discriminator is trained to distinguish between real and fake samples within each domain.

There are a number of works which have adapted this adversarial cycle consistency approach to be applicable across different modalities. Specifically, this has been applied in the field of cross-modal hashing. The task is to create a shared hashing space where data from different modalities can be compared for similarity by comparing their hash.

Li et al. [2] use two separate CycleGANs to learn a cross-modal hash. First, images and text are encoded to feature vectors using a pre-trained convolutional network and word embeddings respectively. Next, the outer CycleGAN learns to translate between the two sets of features. The outer CycleGAN's generator has a bottleneck layer which is yields two latent embeddings–one along each arc of the cycle. A second inner CycleGAN is then trained on to translate between these two latent embeddings. The bottleneck layers of the inner CycleGAN are regularized to match and finally quantized to the final hash representation.

Gu et al. [8] also use a pair of CycleGANs, however, rather than nesting them, they apply them both directly

to the data. One translates from image to text to image (by way of a convolutional encoder and RNN decoder) and the other translates from text to image to text (by way of a RNN encoder and convolutional decoder). Each of these encoder and decoders (4 total) have a feature layer which is marked as the hash code. An adversarial loss is applied to distinguish between the "real" feature layer (encoder) and the "fake" feature layer to enforce consistency between their representations. This approach has two separate embedding spaces, one which is tailored to image-to-text comparison and the other to text-to-image comparison.

While both of these approaches require multiple large CycleGAN networks with difficult adversarial training, they offer a way of learning a multi-modal representation without any explicitly paired data. Such an approach can be used as a submodule within a synthesis network to fuse the representations of disparate input modalities.

4. Case study: synthesizing dance

What has been discussed so far only scratches the surface of the literature relevant to multi-modal synthesis. However, the sheer scale of the possible architectural combinations spanned by these methods is already difficult to reason about. To put the ideas we have seen so far into context, we take the example task of synthesizing dance and will discuss some approaches that try to solve it. This task is a good illustration of many of the aspects we have discussed so far. The input consists of a musical features (a waveform, spectrogram, or features extracted from these) and possibly a prompt of the dancing style. The output is a sequence of two-dimensional or three-dimensional poses which should look like they are dancing. The task is also multi-modal in the sense that there are many plausible dance sequences for a given musical style and dancing style. Ideally, this diversity should be captured by the approach.

4.1. Dancing to Music

The first approach we will discuss is introduced by Lee et al. [49], Dancing to Music. This is a complex, hand-designed architecture with multiple different auto-encoders and GANs pieced together to achieve the desired result. The approach centers around a supervised, multi-stage training framework which the authors call synthesis-by-analysis.

First, dance is analyzed and decomposed into modular "dance units". "Movement beats" (analogous to audio beats) are extracted from training dance sequences and used to segment and normalize dances into parts. A Dance-Unit VAE with two encoders is trained to encode each segment into a pose and movement latent space and then resynthesize the original segment using the two latent vectors.

The next stage (composition) learns to string together the dance units learned in the first stage to synthesis a realistic sequence of poses. This is done with the so-called Music-to-movement GAN. The GAN is trained using music and pose sequences extracted from dance videos. First, the musical style is extracted from the audio and encoded to a dance latent space. The corresponding pose sequence is encoded to the movement latent space and the movement sequence encoded once more to the same "dance space" as the music. A regression loss is placed on the dance latent vectors extracted from the style of the music and pose sequences to enforce the correspondence between music and dance style. Next, both dance latents are fed to a dance de-

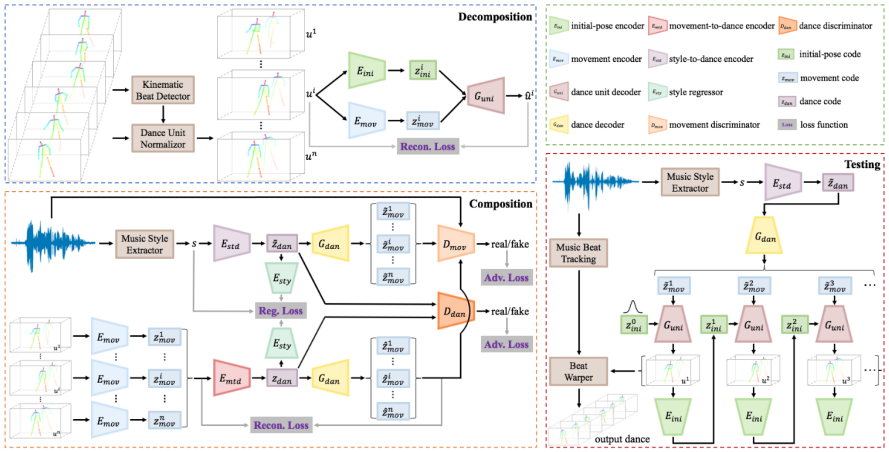


Figure 7. The architecture of Dancing to Music. A complex, hand-designed interplay of multiple networks [49].

coder (generator) to generate a sequence of movement latents again. The dance latents and movement sequences are given to a dance discriminator and movement discriminator for adversarial supervision.

With both stages trained, the style-to-dance encoder, dance decoder, dance-unit decoder, and pose encoder can be composed to synthesize new pose sequences based on unseen audio. Inference works by (1) extracting the musical style and encoding it to dance space, (2) decoding the dance latent to a sequence of movements, (3) recurrently (a) decoding each movement vector to a pose (output sequence) and (b) encoding the pose as the conditioning for the decoding of the next movement vector. Finally, the generated sequence of poses are analyzed for movement beats and warped to match the audio beats.

While the approach is complicated and uses multiple heavy neural networks, it accelled at its task. Dancing to Music significantly improved the previous best approach, MoCoGAN introduced by Tulyakov et al. [50], in terms of both realism, music-style correspondence, and diversity of the synthesized dances. The authors even used a pre-trained Vid2Vid [51] network to translate the generated pose sequences to realistic, high-definition videos of people dancing.

Dancing to Music follows the approach of domain-specific encoders that transform each modality to an initial latent space. Information between the modalities is exchanged by matching encoded music and encoded movement sequences using both a direct regression loss and adversarial supervision. Decoding is finally performed with a hand-designed recurrent architecture that makes use of the generative building blocks that were trained in the analysis phase. This approach takes hand-designed inductive biases to the extreme–making extensive use of expert domain knowledge. While this apparently does work, perhaps this set of networks and tasks are not the easiest to learn via gradient descent. Investigating better architectures in such a large search space of complicated interconnected networks quickly becomes infeasible to do exhaustively. For this reason, a simpler architecture which can perform as well or better would be preferred.

4.2. MoGlow

Henter et al. [52] propose a different architecture that models the pose sequence in a more direct manner; forgoeing multi-stage training and multiple separate latent spaces.

The approach, MoGlow, uses a normalizing flow [53] to model the series of dance poses autoregressively. The normalizing flow consists of a stack of invertible blocks each made up of an affine coupling layer, a linear layer, and activation normalization. The affine coupling layer, first used in the normalizing flow context by Glow [54], learns to modulate its timeseries input using a second conditioning input. In MoGlow this consists of splitting the latent vector in half, transforming the lower half and conditioning information with an LSTM, and finally scale and bias the top half of the latent with the LSTMs outputs. The conditioning information which modulates the normalizing flow's internal LSTM blocks is the concatenation of the previous few poses as well as the corresponding previous audio features. This allows each affine coupling layer to transform a part of the latent to extract or condense some information based on the previous steps of dance and music. The whole process is invertible which means the normalizing flow can perform exact maximumization of the likelihood of the data based on a random, gaussian input.

This architecture directly translates from its input space (poses and music) to its output space (poses) through use of the LSTM in the affine coupling layers. This avoids the complicated auto-encoding pre-training setup used by Lee et al. [49] and can directly optimize its outputs to match the probability density of the target data. There are still a couple drawbacks to using normalizing flows though. They are less expressive than other architectures (often leading to blurrier samples than adversarially trained ones) and are hard to modify for different tasks due to the invertibility constraint.

4.3. AI Choreographer

Another approach which seeks to reduce the amount of inductive biases used to solve the problem is the AI Choreographer (AIC), introduced by Li et al. [55]. They propose a Full-Attention Cross-modal Transformer (FACT) to handle the sequence to sequence modeling. This is a transformer made of two input streams: one for audio features and one for an input pose sequence. The modality-specific transformers encode to a certain latent space which is then concatenated and fed through a final multi-modal transformer which generates the next N steps in the dance. The audio transformer takes in features of the music including onsets, MFCCs, and the chromagram while the pose transformer takes in the pose sequence directly. The authors find that an early fusion setup seems to help the output dance vary more strongly with the musical style. They elect for only 2 transformer blocks for the input transformers and 12 blocks for the following multi-modal section. They further note that autoregressive, future-N generation is crucial to prevent freezing or jerky movements seen in previous work.

This approach directly follows some of the other

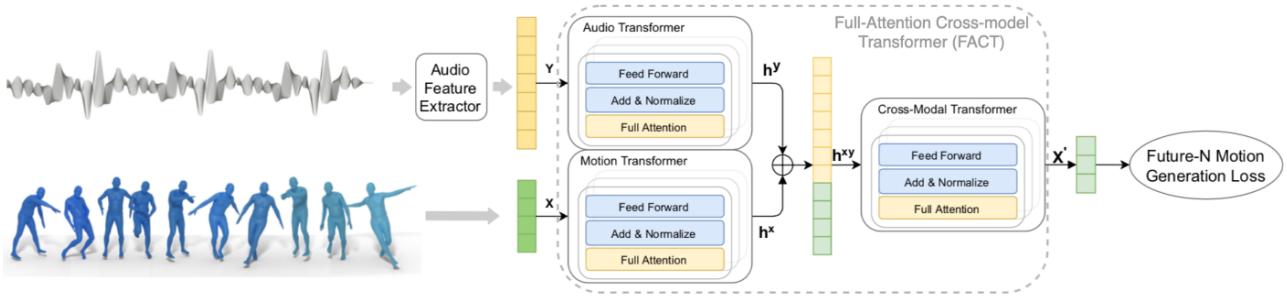


Figure 8. The architecture of AI Choreographer. A multi-modal, middle-fusion transformer [55].

attention-based approaches we discussed in subsection 3.2. Some domain knowledge is included in the feature extraction used as input to the audio transformer, however, this could be reduced by using a Perceiver-based encoder to ingest the audio waveform directly. The rest of the architecture is fairly devoiod of hand-coded inductive biases. The transformer is used to find which correlations in the data are most important for the task at hand. Being a sequence to sequence task, there is also no need for a specific decoder as the pose sequence can be output by the transformer directly.

One major drawback with AIC, though, is the lack of probabilistic modeling of the dance sequences. While the output can be changed by differing the pose sequence input, the model is unable to generate different plausible sequences from a fixed input.

4.4. Transflower

Valle-Pérez et al. [56] seek to combine the best aspects of AIC and MoGlow; they simply concatenate the two architectures. The outputs of the FACT network from AIC are treated as a latent sequence and rather used as conditioning information for the normalizing flow from MoGlow. They further replaces the linear layer in the flow blocks with invertible 1x1 convolutions and activation normalization with batch normalization. They also use a different positional encoding in the FACT, rather opting for relative positional encoding as introduced by T5 [57]. While there are some architectural changes, the basis remains the same. This architecture merges the best aspects of both models allowing for the high-quality musical reactivity from AIC as well as the explicit probabilistic modeling from MoGlow.

It is interesting to note in the field of dance synthesis the state-of-the-art has progressed towards architectures which have less explicitly-coded inductive biases. Rather than using modality-specific encoders and bridging between different latent spaces hand-designed to represent specific aspects of the problem, the newer approaches use general architectures which can directly transform input data to the desired output. This highlights the importance of understanding the approaches that divide responsibilities between different network modules and how the same functionality can be offered by models which are able to encode the information wholistically.

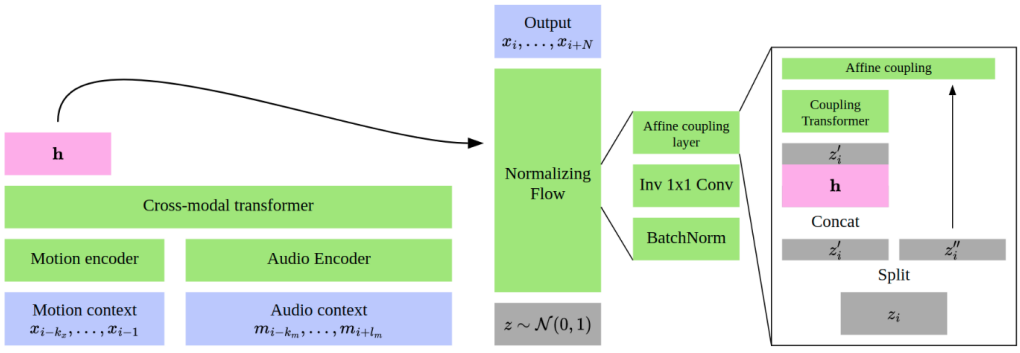


Figure 9. The architecture of Transflower. A combination of the cross-modal transformer and Glow network [56].

5. Conclusion

Multi-modal data synthesis is a broad field which draws from a large variety of different sub-fields. Even so, there are certain general trends in the approaches that are adopted to solve problems, regardless of the specific modalities being handled. Dividing the multi-modal synthesis process into encoding/decoding of modalities and exchanging information across modalities, we see that there are a few archetypes which most methods fall into. Many approaches make use of modality-specific submodules which encode different modalities to condensed intermediate representations. The corresponding cross-modal information integration then generally consists of trying to unify these representations with losses specified between the embeddings of related data. In a synthesis context, these shared representations are often then decoded back to one of the original modalities and supervised via a reconstruction loss. However, recently there has also been a trend towards methods which do not require specific encoders/decoders or latent spaces for specific aspects of the task. This is often achieved through the use of attention-based approaches such as transformers. These approaches benefit from much simpler overall architectures with less explicitly designed inductive biases. These models can learn on their own which aspects of the inputs should be combined and how this can best be done towards a specific goal.

Within both modality-specifc and more modality-agnostic methods, there is still much reliance on explicit supervision and paired data. Being able to loosen this requirement is an important direction for continued research. The literature has already layed out multiple strategies for reducing certain data pairings in specific situations, for example: completely unpaired in the case of cycle consistency approaches, unpaired input/output modalities via an intermediate paired modality, or unpaired example data by swapping pre-trained networks. Generalizing these approaches to span across multiple modalities and broaden their applicability can greatly reduce data-gathering requirements for multi-modal synthesis problems. Contrastive learning offers another great avenue towards less explicit supervision in training methods. Loosening positive sample pairings through clever augmentation schemes that are compatible with multiple modalities can help reduce the need for explicit paired data as well.

Both model architectures and training requirements are converging towards a framework that is completely ambivalent to the types of data that are used as inputs and outputs. Achieving such a feat will still require quite some innovation on all sides. However, the current strides being made in multi-modal synthesis even on large-scale high-definition modalities such as video or audio are heartening. There are already many solutions that can create convincing, evocative novel data– especially when there is ample paired data to train on.

References

[1] Chao Li, Cheng Deng, Ning Li, Wei Liu, Xinbo Gao, and Dacheng Tao. Self-Supervised Adversarial Hashing Networks for Cross-Modal Retrieval. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4242–4251, . doi: 10.1109/CVPR.2018.00446.

[2] Chao Li, Cheng Deng, Lei Wang, De Xie, and Xianglong Liu. Coupled CycleGAN: Unsupervised Hashing Network for Cross-Modal Retrieval, . URL `http://arxiv.org/abs/1903.02149`.

[3] De Xie, Cheng Deng, Chao Li, Xianglong Liu, and Dacheng Tao. Multi-Task Consistency-Preserving Adversarial Hashing for Cross-Modal Retrieval. URL `https://ieeexplore.ieee.org/document/8954946`.

[4] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver IO: A General Architecture for Structured Inputs & Outputs, . URL `http://arxiv.org/abs/2107.14795`.

[5] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General Perception with Iterative Attention, . URL `http://arxiv.org/abs/2103.03206`.

[6] Dan Li, Changde Du, and Huiguang He. Semi-supervised cross-modal image generation with generative adversarial networks. 100:107085, . ISSN 0031-3203. doi: 10.1016/j.patcog.2019.107085. URL `https://www.sciencedirect.com/science/article/pii/S0031320319303863`.

[7] Yuxin Peng and Jinwei Qi. CM-GANs: Cross-modal Generative Adversarial Networks for Common Representation Learning. 15(1):22:1–22:24. ISSN 1551-6857. doi: 10.1145/3284750. URL `https://doi.org/10.1145/3284750`.

[8] Jingzi Gu, Peng Fu, Jinchao Zhang, Lulu Wang, Bo Li, and Weiping Wang. Unsupervised Cross-Modal Retrieval by Coupled Dual Generative Adversarial Networks. In Xin Wang, Rui Zhang, Young-Koo Lee, Le Sun, and Yang-Sae Moon, editors, *Web and Big Data*, Lecture Notes in Computer Science, pages 574–587. Springer Interna-

tional Publishing. ISBN 978-3-030-60259-8. doi: 10.1007/978-3-030-60259-8_42.

[9] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the Devil in the Details: Delving Deep into Convolutional Nets. URL http://arxiv.org/abs/1405.3531.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, . doi: 10.1109/CVPR.2009.5206848.

[11] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. URL http://arxiv.org/abs/1409.1259.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. URL http://arxiv.org/abs/1810.04805.

[13] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. URL http://arxiv.org/abs/1406.2661.

[14] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. URL http://arxiv.org/abs/1411.1784.

[15] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. URL http://arxiv.org/abs/1312.6114.

[16] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N. Metaxas, and Sergey Tulyakov. A Good Image Generator Is What You Need for High-Resolution Video Synthesis. URL http://arxiv.org/abs/2104.15069.

[17] Gereon Fox, Ayush Tewari, Mohamed Elgharib, and Christian Theobalt. StyleVideoGAN: A Temporal Generative Model using a Pretrained StyleGAN. URL http://arxiv.org/abs/2107.07224.

[18] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. URL http://arxiv.org/abs/1912.04958.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. 9(8):1735–1780. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

[20] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in Style: A StyleGAN Encoder for Image-to-Image Translation. URL http://arxiv.org/abs/2008.00951.

[21] Justin N. M. Pinkney and Doron Adler. Resolution Dependent GAN Interpolation for Controllable Image Synthesis Between Domains. URL http://arxiv.org/abs/2010.05334.

[22] NVlabs/ffhq-dataset. URL https://github.com/NVlabs/ffhq-dataset.

[23] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse Image Synthesis for Multiple Domains. URL http://arxiv.org/abs/1912.01865.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. URL http://arxiv.org/abs/1706.03762.

[25] Jianqiao Zheng, Sameera Ramasinghe, and Simon Lucey. Rethinking Positional Encoding. URL http://arxiv.org/abs/2107.02561.

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. URL http://arxiv.org/abs/2103.00020.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. doi: 10.1109/CVPR.2016.90.

[28] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. URL http://arxiv.org/abs/2010.11929.

[29] CLIP. URL https://github.com/openai/CLIP.

[30] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. VATT: Transformers for Multimodal Self-Supervised Learning from Raw Video, Audio and Text. URL http://arxiv.org/abs/2104.11178.

[31] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention Bottlenecks for Multimodal Fusion. URL http://arxiv.org/abs/2107.00135.

[32] Jort F. Gemmeke, Daniel P. W. 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 *Proc. IEEE ICASSP 2017*.

[33] Michael Sollami and Aashish Jain. Multimodal Conditionality for Natural Language Generation. URL http://arxiv.org/abs/2109.01229.

[34] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1. doi: 10.1109/CVPR.2005.202.

[35] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304. JMLR Workshop and Conference Proceedings. URL https://proceedings.mlr.press/v9/gutmann10a.html.

[36] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. URL http://arxiv.org/abs/1807.03748.

[37] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. URL http://arxiv.org/abs/1912.06430.

[38] Sangho Lee, Jiwan Chung, Youngjae Yu, Gunhee Kim, Thomas Breuel, Gal Chechik, and Yale Song. ACAV100M: Automatic Curation of Large-Scale Datasets for Audio-Visual Video Representation Learning, . URL http://arxiv.org/abs/2101.10803.

[39] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast Networks for Video Recognition. URL http://arxiv.org/abs/1812.03982.

[40] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron Weiss, and Kevin Wilson. CNN Architectures for Large-Scale Audio Classification. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. URL https://arxiv.org/abs/1609.09430.

[41] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive Learning for Unpaired Image-to-Image Translation. URL http://arxiv.org/abs/2007.15651.

[42] Mandela Patrick, Yuki M. Asano, Polina Kuznetsova, Ruth Fong, João F. Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal Self-Supervision from Generalized Data Transformations. URL http://arxiv.org/abs/2003.04298.

[43] Vikas Verma, Minh-Thang Luong, Kenji Kawaguchi, Hieu Pham, and Quoc V. Le. Towards Domain-Agnostic Contrastive Learning. URL http://arxiv.org/abs/2011.04419.

[44] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. Mixup: Beyond Empirical Risk Minimization. URL https://arxiv.org/abs/1710.09412v2.

[45] Jing Gao, Wenjun Zhang, Fangming Zhong, and Zhikui Chen. UCMH: Unpaired cross-modal hashing with matrix factorization. 418:178–190. ISSN 0925-2312. doi: 10.1016/j.neucom.2020.08.029. URL https://www.sciencedirect.com/science/article/pii/S0925231220312923.

[46] Shuang Ma, Daniel McDuff, and Yale Song. Unpaired Image-to-Speech Synthesis with Multimodal Information Bottleneck. URL http://arxiv.org/abs/1908.07094.

[47] Kangle Deng, Aayush Bansal, and Deva Ramanan. Unsupervised Audiovisual Synthesis via Exemplar Autoencoders, . URL http://arxiv.org/abs/2001.04463.

[48] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. URL http://arxiv.org/abs/1703.10593.

[49] Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. Dancing to Music, . URL http://arxiv.org/abs/1911.02001.

[50] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing Motion and Content for Video Generation. URL http://arxiv.org/abs/1707.04993.

[51] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-Video Synthesis. URL https://arxiv.org/abs/1808.06601v2.

[52] Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. MoGlow: Probabilistic and controllable motion synthesis using normalising flows. 39(6): 1–14. ISSN 0730-0301, 1557-7368. doi: 10.1145/3414685.3417836. URL http://arxiv.org/abs/1905.06598.

[53] Ivan Kobyzev, Simon J. D. Prince, and Marcus A. Brubaker. Normalizing Flows: An Introduction and Review of Current Methods. doi: 10.1109/TPAMI.2020.2992934. URL https://arxiv.org/abs/1908.09257v4.

[54] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative Flow with Invertible 1x1 Convolutions. URL https://arxiv.org/abs/1807.03039v2.

[55] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. AI Choreographer: Music Conditioned 3D Dance Generation with AIST++, . URL http://arxiv.org/abs/2101.08779.

[56] Guillermo Valle-Pérez, Gustav Eje Henter, Jonas Beskow, André Holzapfel, Pierre-Yves Oudeyer, and Simon Alexanderson. Transflower: Probabilistic autoregressive dance generation with multimodal attention. URL http://arxiv.org/abs/2106.13871.

[57] 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. URL http://arxiv.org/abs/1910.10683.