Regeneration Learning: A Learning Paradigm for Data Generation

Xu Tan[1], **Tao Qin**[1], **Jiang Bian**[1], **Tie-Yan Liu**[1], **Yoshua Bengio**[2,3]

[1]Microsoft Research
[2]Mila
[3]University of Montreal
[1]{xuta,taoqin,jiabia,tyliu}@microsoft.com, [2]yoshua.bengio@mila.quebec

Abstract

Machine learning methods for conditional data generation usually build a mapping from source conditional data X to target data Y. The target Y (e.g., text, speech, music, image, video) is usually high-dimensional and complex, and contains information that does not exist in source data, which hinders effective and efficient learning on the source-target mapping. In this paper, we present a learning paradigm called *regeneration learning* for data generation, which first generates Y' (an abstraction/representation of Y) from X and then generates Y from Y'. During training, Y' is obtained from Y through either handcrafted rules or self-supervised learning and is used to learn $X \rightarrow Y'$ and $Y' \rightarrow Y$. Regeneration learning extends the concept of representation learning to data generation tasks, and can be regarded as a *counterpart* of traditional representation learning, since 1) regeneration learning handles the abstraction (Y') of the target data Y for data generation while traditional representation learning handles the abstraction (X') of source data X for data understanding; 2) both the processes of $Y' \rightarrow Y$ in regeneration learning and $X \rightarrow X'$ in representation learning can be learned in a self-supervised way (e.g., pre-training); 3) both the mappings from X to Y' in regeneration learning and from X' to Y in representation learning are simpler than the direct mapping from X to Y. We show that regeneration learning can be a widely-used paradigm for data generation (e.g., text generation, speech recognition, speech synthesis, music composition, image generation, and video generation) and can provide valuable insights into developing data generation methods.

Introduction

Data Understanding and Generation

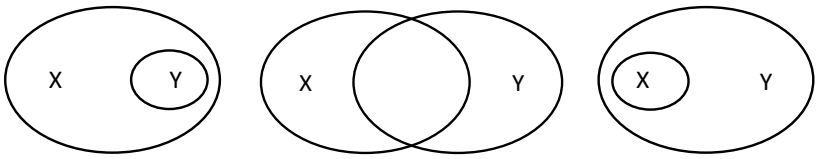
Typical machine learning tasks, in the field of natural language processing (Manning and Schutze 1999; Collobert et al. 2011; Vaswani et al. 2017; Devlin et al. 2018; Brown et al. 2020), speech (Benesty et al. 2008; Hinton et al. 2012; Oord et al. 2016; Wang et al. 2017; Tan et al. 2022), computer vision (Forsyth and Ponce 2011; Szeliski 2010; Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016; Goodfellow et al. 2014), and etc, usually handle a mapping from source data X to target data Y. For example, X is image and Y is class label in image classification (Deng et al. 2009); X is style tag and Y is sentence in style-controlled text generation (McKeown 1992); X is text and Y is speech in text-to-speech synthesis (Tan et al. 2021, 2022).

Depending on the relative amount of information that X and Y contain, these mappings can be divided into data understanding (Krizhevsky, Sutskever, and Hinton 2012; Devlin et al. 2018), data generation (Goodfellow et al. 2014; Brown et al. 2020), and the combination of data understanding and generation (Bahdanau, Cho, and Bengio 2014; Hassan et al. 2018; Graves 2012; Chan et al. 2016; Tan et al. 2022). Figure 1 shows the three types of tasks and the relative information between X and Y:

- Data understanding tasks, in which X contains much more information than Y (e.g., image classification (Deng et al. 2009; Krizhevsky, Sutskever, and Hinton 2012), objective detection (Girshick 2015; Redmon et al. 2016), sentence classification (Zhang and Wallace 2015), machine reading comprehension (Rajpurkar et al. 2016)).
- Data generation tasks, in which Y contains much more information than X (e.g., text generation (Brown et al. 2020) or image synthesis (Goodfellow et al. 2014; Kingma and Welling 2013) from class label).
- Data understanding/generation tasks, in which X contains no significantly more or less information than Y (e.g., image transfer (Zhu et al. 2017), text-to-image synthesis (Ramesh et al. 2021, 2022; Yu et al. 2022; Saharia et al. 2022; Chang et al. 2023), neural machine translation (Bahdanau, Cho, and Bengio 2014; Hassan et al. 2018), text-to-speech synthesis (Tan et al. 2022, 2021), automatic speech recognition (Hinton et al. 2012)). In this case, we need both data understanding capability on the source X and data generation capability on the target Y.

The information mismatch between X and Y leads to different strategies for solving different tasks. For data understanding tasks, X is usually high-dimensional, complex, and redundant compared to Y, and the key is to learn highly abstractive or discriminative representations (sometimes need to remove unnecessary information) for X in order to better predict Y. Thus, representation learning (Bengio, Courville, and Vincent 2013)[1] and especially self-

[1]One of the most impactful conferences in deep learning is ICLR, which is short for "The International Conference on Learning Representations".



(a) Data understanding (b) Data understanding/generation (c) Data generation

Types	Information	Tasks
Understanding	$X \gg Y$	image classification, objective detection sentence classification, reading comprehension
Generation	$X \ll Y$	text generation or image synthesis from ID/class
Understanding/Generation	$X \not\gg Y$ and $X \not\ll Y$	text to speech, automatic speech recognition, text to image generation, talking-head synthesis

Figure 1: Three types of tasks in machine learning and the relative information between source X and target Y.

supervised pre-training (Devlin et al. 2018; Brown et al. 2020) have become some of the hottest topics in deep learning research in the past years. For data generation tasks, Y is usually high-dimensional, complex, and redundant compared to X, and the key is how to better represent the distribution of Y and better generate Y from X. For data understanding/generation tasks, they need the capability in both understanding and generation, i.e., extract good representations from X and fully generate the information in Y.

Challenges of Data Generation

For the data generation tasks and the generation part of the data understanding/generation tasks (we call both the two types as data generation tasks in the remaining of this paper), they face distinctive challenges that cannot be addressed by the traditional formulation of representation learning.

- First, for the generation tasks where Y contains much more complex information than X, the generation models face severe one-to-many mapping problems (ill-posed or ill-condition problem) (Bertero, Poggio, and Torre 1988), which increases the learning difficulty. For example, in class-conditioned image generation, a class label "dog" can correspond to different images that contain dogs. Incorrect modeling on the ill-posed problem could result in overfitting on the training set and poor generalization on the test set.
- Second, for the generation tasks (e.g., speech recognition (Hinton et al. 2012), speech synthesis (Tan et al. 2021), talking-head video synthesis (Thies et al. 2020)) where X contains no significant more or less information than Y, there are two situations: 1) the mapping between X and Y is not one-to-one (e.g., multiple words with the same pronunciation can correspond to one speech segment in automatic speech recognition, and multiple speech with different speaking rates can correspond to one text sequence in text-to-speech synthesis), which faces the same problem mentioned above; 2) there are some spurious correlations between X and Y, e.g., the speaking timber in source speech has no correlation with the head pose in target video (Chen et al. 2019) for talking-head video synthesis, and some information in target melody has no correlation with source lyric in lyric-to-melody generation (Ju et al. 2021). Fitting these spurious correlations can be harmful for generation in inference.

Why Regeneration Learning

Some generative models (e.g., GANs (Goodfellow et al. 2014), VAEs (Kingma and Welling 2013), autoregressive models (Oord et al. 2016; Brown et al. 2020), normalizing flows (Rezende and Mohamed 2015; Kingma and Dhariwal 2018), diffusion models (Ho, Jain, and Abbeel 2020)) have achieved rapid progress in a variety of data generation tasks. Ideally, as long as generative models are powerful enough, they can fit any complex data distribution. However, in practice, they cannot model the complex distribution and one-to-many mapping well due to many reasons, such as too complicated data mapping, too heavy computation cost, and data sparsity, etc. In analogy to data understanding tasks, although learning a powerful model (e.g., CNN or Transformer) to directly classify source data into target labels would ideally achieve very good accuracy, it still suffers from low accuracy, and some advanced representation learning methods such as large-scale self-supervised pre-training (Devlin et al. 2018) can greatly boost the accuracy.

In this paper, we present a learning paradigm called *regeneration learning* for data generation tasks. Instead of directly generating target data Y from source data X, regeneration learning first generates Y' (a representation of Y) from X and then generates Y from Y'. Regeneration learning extends the concept of representation learning to data generation tasks and learns a good representation (Y') of the target data Y to ease the generation: 1) $X \to Y'$ mapping will be less one-to-many than $X \to Y$ since Y' is a compact/representative version of Y; 2) $Y' \to Y$ mapping can be learned in a self-supervised way (Y' is obtained from Y) and can be empowered by large-scale pre-training that is similar to that in traditional representation learning for data understanding tasks (e.g., BERT (Devlin et al. 2018)).

In the rest of this paper, we first introduce the basic formulation of regeneration learning and its connection to other

Formulation	Category	Method	Data Conversion ($Y \rightarrow Y'$)
Basic	Explicit	Fourier Transformation	Speech/Image (e.g, Wave→ Spectrogram)
		Grapheme-to-Phoneme	Text (e.g., learning→ˈlɜːrnɪŋ)
		Music Analysis	Music (MIDI→Chord/Rhythm)
		3D Image Analysis	Image (Face to 3D Co-efficient)
		Down Sampling	Speech/Image (e.g., 256*256→64*64)
	Implicit	Analysis-by-Synthesis VAE VQ-VAE/VQ-GAN DiffusionAE	Image/Speech/Text ($Y \rightarrow Z$)
Extended	Factorization	AR	Image/Speech/Text ($Y \rightarrow Y_{1:t}$)
	Diffusion	DDPM	Image/Speech/Text ($Y_0 \rightarrow Y_t$)
	Latent Diffusion	VAE + DDPM	Image/Speech/Text ($Y \rightarrow Z_0$, $Z_0 \rightarrow Z_t$)

Table 1: Different methods for $Y \rightarrow Y'$ conversion.

learning methods and paradigms in Section , then summarize the applications of regeneration learning in Section , and finally list some research opportunities on regeneration learning in Section .

Formulations of Regeneration Learning

In this section, we introduce regeneration learning, which leverages intermediate representations of target data Y to bridge the information mismatch between X and Y. There are three steps in regeneration learning:

- Step 1: Convert target data Y into an abstractive/representative version Y'.
- Step 2: Learn a model to generate Y' from source data X.
- Step 3: Learn another model to generate Y from Y'.

This learning paradigm is called *regeneration learning* due to the following reasons: 1) Literally, it has two generation steps that first generate Y' and then generate Y (i.e., regenerate), which is in analogy to what "represent" is to "present" in representation learning[2]. 2) Metaphorically, in analogy to what "represent" in representation learning means, i.e., "using one thing to signify another thing", the word "regenerate" in regeneration learning means to generate one thing to signify another thing (i.e., generate Y' to signify Y). Regeneration learning has several advantages: 1) the mapping from Y' to Y can be learned in a self-supervised way, which is much more data-efficient; 2) the mapping from X to Y' is simpler than the direct mapping from X to Y.

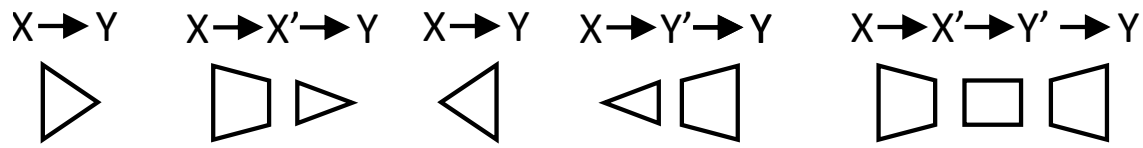
We first introduce the three steps in regeneration learning in section and discuss the connections of regeneration learning to existing methods in Section and the relationships between regeneration learning and representation learning in Section .

[2]Strictly speaking, "represent" in representation learning means "using one thing to signify another thing", which is different from "re-present" that means "to present again". However, the meaning of "represent" has a close relation to "re-represent": suppose you use X' to signify X (i.e., represent X with X' and thus X' is the representation of X), and then in this case, you actually re-present X using X'.

Basic Formulation

Extract Y' from Target Y. There are three principles when converting Y to Y': 1) Y' should be more abstractive and representative than Y; 2) the removed information from Y to Y' has no or little correlation with source data X, i.e., Y' is a compact version of Y but still maintains its correlation with X; 3) the conversion from Y to Y' should be easy, e.g., processed by simple transformation or extraction tools, or at least not relying on labeled data if model learning is needed. According to the above principles, there are different ways to convert Y into Y', as shown in the "Basic Formulation" in Table 1:

- Explicit transformation. We can convert Y into Y' with some explicit transformation methods: 1) Mathematical transformation such as Fourier or Wavelet transformation. For example, we can convert a speech waveform into a sequence of linear-scale or mel-scale spectrograms using short-time Fourier transformation (STFT) (Wang et al. 2017; Shen et al. 2018). 2) Modality transformation such as grapheme-to-phoneme conversion. For example, we can convert a text/character/grapheme sequence into a phoneme sequence using the grapheme-to-phoneme conversion model (Sun et al. 2019). 3) Data analysis. For example, we can extract music templates (chord, rhythm, etc) from a melody sequence using some music analysis tools to get the abstraction of the music (Ju et al. 2021) or extract the 3D face parameters from a face image (Ling et al. 2022) using a 3D face model (Blanz and Vetter 1999). 4) Downsampling. For example, we can simply down-sample an image from $256 * 256$ resolution to $64 * 64$ or a speech sequence from 48kHz sampling rate to 24kHz. These transformation methods are usually built on well-established methods or tools, and the Y' transformed from Y is usually in an explicit data format.
- Implicit transformation. Different from explicit transformation that converts Y into Y' with explicit data format using some well-built rules, algorithms, or models, end-to-end learning achieves this by learning an intermediate and implicit representation through analysis-by-synthesis pipeline or reconstruction. Some commonly used models include auto-encoder (AE), denoising auto-



(a) (b) (c) (d) (e)

Figure (a): Presentation ($X \rightarrow$). Figure (b): Representation ($X \rightarrow X' \rightarrow$). Figure (c): Generation ($\rightarrow Y$). Figure (d): Regeneration ($\rightarrow Y' \rightarrow Y$). Figure (e): Representation + Regeneration ($X \rightarrow X' \rightarrow Y' \rightarrow Y$).

Paradigm	Original	Compact	Self-Supervised Learning	Easy Mapping
(b) Representation Learning	X	X'	$X \rightarrow X'$	$X' \rightarrow Y$
(d) Regeneration Learning	Y	Y'	$Y' \rightarrow Y$	$X \rightarrow Y'$
(e) Combination	X, Y	X', Y'	$X \rightarrow X', Y' \rightarrow Y$	$X' \rightarrow Y'$

Figure 2: Comparison between regeneration learning and traditional representation learning on $X \rightarrow Y$. The triangle, trapezoid, and quadrangle in Figure (a)-(e) represent the information changes. Figure (a): Data understanding tasks, where X contains more information than Y. Figure (b): Data understanding tasks with representation learning, where X' is a more compact version of X. Figure (c): Data generation tasks, where Y contains more information than X. Figure (d): Data generation tasks with regeneration learning, where Y' is a more compact version of Y. Figure (e): Data understanding/generation tasks with the combination of representation and regeneration learning (e.g., sequence-to-sequence learning tasks such as speech to talking-head video synthesis), where X' and Y' contain a comparable amount of information and are compact versions of X and Y respectively (e.g., X' could be speech representations learned by self-supervised models, and Y' could be 3D coefficients extracted by 3D face model in talking-head video synthesis).

encoder (DAE), variational auto-encoder (VAE) (Kingma and Welling 2013), vector-quantized auto-encoder (VQ-VAE) (van den Oord, Vinyals, and Kavukcuoglu 2017; Razavi, van den Oord, and Vinyals 2019), etc. Beyond the learned encoder that converts Y into Y', they can additionally learn a decoder to convert Y' back to Y, which can be used in the third step of regeneration learning.

Generate Y' from Source X. The generation from X to Y' can be approached by any machine learning method, similar to those used to model $X \rightarrow Y$ (e.g., Autoregressive Model, GAN, VAE, Flow, Diffusion). Since Y' is extracted from Y satisfying the above three principles, the task of $X \rightarrow Y'$ is easier than that of $X \rightarrow Y$. For example, generating mel-spectrograms from text is much simpler than generating waveform in text-to-speech synthesis, generating 3D face parameters from the speech is much simpler than generating face image in talking-head video synthesis, and generating representation sequence (latent code sequence) from source condition is much simpler than generating pixel-level images in the conditional image or video generation.

Generate Target Y from Y'. Since we can easily extract Y' from Y by tools or models without relying on any paired training data (i.e., X and Y), we can train a model to predict Y from Y' in a self-supervised way, where the paired training data (Y', Y) can be collected in large scale without much cost. We can also conduct pre-training to enhance the capability of $Y' \rightarrow Y$ by only using a large amount of unpaired data Y. Note that when using auto-encoding methods to convert Y into Y', we can already get a decoder that converts Y' to Y, without the need to train another model.

Infer Y from X via Y'. We discuss how to combine the two generation steps $X \rightarrow Y'$ and $Y' \rightarrow Y$ together to generate Y from X. Note that since $X \rightarrow Y$ mapping is one-to-many, the prediction of Y from X is distribution-wise, i.e., $Y \sim P(Y|X)$, instead of point-wise. Similarly, since Y' is a compact version of Y that incurs information loss, $Y' \rightarrow Y$ is also a one-to-many mapping and should also be modeled in a distribution-wise way, i.e., $Y \sim P(Y|Y')$. Furthermore, although the ill-posed mapping problem in $X \rightarrow Y'$ is largely alleviated compared to that in $X \rightarrow Y$, the $X \rightarrow Y'$ mapping is still one-to-many in general, and thus should be modeled in a distribution-wise way too, i.e., $Y' \sim P(Y'|X)$. On the other hand, $Y \rightarrow Y'$ is usually point-wise since it is converted by a deterministic function. But it can also be distribution-wise, such as using a VAE encoder to get the mean and variance of Y' from Y. However, no matter whether $Y \rightarrow Y'$ is point-wise or distribution-wise, $X \rightarrow Y'$ and $Y' \rightarrow Y$ should be distribution-wise. We usually leverage deep generative models (e.g., autoregressive models, GANs, VAEs, normalizing flows, and denoising diffusion probabilistic models) to learn the conditional distributions $P(Y'|X)$ and $P(Y|Y')$. After that, given a data sample X, we can first sample Y' from the conditional distribution $P(Y'|X)$, and then given the sampled data Y', we can further sample Y from the conditional distribution $P(Y|Y')$, i.e., $Y' \sim P(Y'|X)$, $Y \sim P(Y|Y')$.

Connections to Other Methods

We list some methods that have connections to regeneration learning, including the methods that are basic and extended versions of regeneration learning, and that do not belong but are related to regeneration learning, as shown in Table 1.

Regeneration vs. Representation Learning

Regeneration learning extends the concept of representation learning to data generation, and thus it can be regarded as a

Task	X	Y	Y'	$Y \rightarrow Y'$ & $Y' \rightarrow Y$
Speech Synthesis	Text	Waveform	Spectrogram / Code	STFT & Vocoder / Codec
Speech Recognition	Speech	Character	Phoneme	G2P & P2G
Text Generation	Text/Knowledge	Text	Template	Text2Template & Template2Text
Lyric/Video to Melody	Lyric/Video	Melody	Music Template	Music Analysis & Generation
Talking-Head Synthesis	Speech	Video	3D Face Parameters	3D Face Analysis & Rendering
Image/Video/Sound Generation	Class/Text	Image/Video/Sound	Latent Code	Codec Extraction & Generation

Table 2: Typical data (text, speech, music, sound, image, and video) generation tasks that leverage regeneration learning.

special type of representation learning for data generation. Furthermore, we can regard regeneration learning as a *counterpart* of traditional representation learning, since 1) regeneration learning handles the abstraction (Y') of the target data Y for data generation, while traditional representation learning handles the abstraction (X') of source data X for data understanding; 2) both the processes of $Y' \rightarrow Y$ in regeneration learning and $X \rightarrow X'$ in traditional representation learning can be learned in a self-supervised way (e.g., pre-training); 3) both the mappings from X to Y' in regeneration learning and from X' to Y in traditional representation learning are simpler than the direct mapping from X to Y. Figure 2 shows the comparison between regeneration learning and traditional representation learning.

Applications of Regeneration Learning

A variety of tasks in conditional data generation (e.g., text generation, speech recognition, speech synthesis, music composition, image generation, and video generation) can benefit from this regeneration learning paradigm. We list some typical generation tasks in Table 2.

Basically speaking, a conditional data generation task can leverage regeneration learning as long as they fit into some situations:

- The target data is too high-dimensional and complex to generate, or incurs too much computation cost, such as waveform generation in text-to-speech synthesis (Shen et al. 2018; Ren et al. 2019; Tan et al. 2021), and image/video/sound generation (Ramesh et al. 2021; Ding et al. 2021; Yan et al. 2021; Rakhimov et al. 2020; Rombach et al. 2022; Kreuk et al. 2022). In this case, converting target data Y into more compact Y' will greatly reduce the computation cost and free the model to focus more on how to generate high-level abstractive or semantic information of target data, but not on the minor details.
- The source data X and target data Y have too much uncorrelated information (i.e., $X \cap Y \ll X \cup Y$), such as lyric/video and melody in conditional melody generation (Ju et al. 2021; Wu et al. 2020; Dai et al. 2021; Zou et al. 2021; Di et al. 2021), speech and face images in talking-head video synthesis (Thies et al. 2020; Ji et al. 2021; Yi et al. 2020; Lahiri et al. 2021; Song et al. 2021; Ling et al. 2022). Directly learning the mapping between X and Y would lead to overfitting. Thus, converting Y into more compact Y' will make the mapping between X and Y' less ill-posed and ease the model learning.
- There lack of paired X and Y, and thus regeneration learning can be leveraged to train $Y' \rightarrow Y$ with large-scale self-supervised learning based on only target data Y.

Opportunities on Regeneration Learning

We discuss some research opportunities to make regeneration learning more powerful to solve a variety of data generation tasks, mainly from three perspectives: 1) how to get Y'; 2) how to learn the mapping $X \rightarrow Y'$ and $Y' \rightarrow Y$; 3) how to reduce the training-inference mismatch in regeneration learning pipeline.

How to Get Y'

How to find an appropriate Y' is important for $X \rightarrow Y'$ and $Y' \rightarrow Y$ mapping. For example, in text-to-speech synthesis, mel-spectrograms and mel-frequency cepstral coefficients (MFCCs) are both possible Y' for target waveform Y. However, mel-spectrograms are demonstrated to be much better than MFCCs, since MFCCs are too abstractive that lose a lot of fine-grained information, thus making it difficult to reconstruct Y from Y'. In the following, we list several possible research points to get a better Y'.

Better Implicit Learning. Beyond using some handcrafted rules or well-developed tools to find Y', we can automatically learn Y' from Y. The typical methods are based on an analysis-by-synthesis pipeline, e.g., variational auto-encoder (VAE) (Kingma and Welling 2013) and vector-quantized variational auto-encoder (VQ-VAE) (van den Oord, Vinyals, and Kavukcuoglu 2017). For example, in text-to-speech synthesis (Cong et al. 2021; Tan et al. 2022; Liu et al. 2022), NaturalSpeech (Tan et al. 2022) leverages VAE and DelightfulTTS 2 (Liu et al. 2022) leverages VQ-VAE to learn intermediate representations Y' from speech waveform Y and reconstruct Y from Y'. The motivation is that the commonly-used intermediate representations (e.g., mel-spectrograms) are extracted by SFTF algorithms that may not be optimal, while those learned by VAE could be better representations of waveform and ease the generation process. In the image and video domain, VQ-VAE (van den Oord, Vinyals, and Kavukcuoglu 2017) is widely used to learn discrete visual tokens to represent images and videos. However, some other works try to further improve the discrete token extraction of VQ-VAE by introducing hierarchical learning (e.g., VQ-VAE-2 (Razavi, van den Oord, and Vinyals 2019)), adding adversarial loss (e.g., VQ-GAN (Esser, Rombach, and Ommer 2021)) and perceptual loss (e.g., PECO (Dong et al. 2021)), residual quantizers (Gray 1984; Zeghidour et al. 2021; Défossez et al. 2022), or leverage diffusion models to learn the hidden representation (Preechakul et al. 2022).

- **RQ1**: How to design better analysis-by-synthesis methods (beyond VAE, VQ-VAE, DiffusionAE, etc.) to learn Y'?
- **RQ2**: How to design better learning paradigms other than analysis-by-synthesis to learn Y'?

Learning Y' by Considering X. An intuition is that the abstractive representation Y' should not only depends on Y, but also be correlated to X in order to facilitate the prediction of Y' from X. Besides, there is a trade-off on the difficulties between $X \rightarrow Y'$ and $Y' \rightarrow Y$, since considering more X when learning Y' will ease the learning of $X \rightarrow Y'$ while making the learning of $Y' \rightarrow Y$ harder. We should make a good trade-off to get better overall performance.

- **RQ3**: How to leverage unpaired data Y and/or paired data (X, Y) to learn Y'?
- **RQ4**: How to better trade off the difficulty between $X \rightarrow Y'$ and $Y' \rightarrow Y$ mappings when learning Y'?

Semantic and Perceptual Disentanglement. Generally speaking, $X \rightarrow Y'$ cares more about semantic mapping/-conversion from source to target, while $Y' \rightarrow Y$ cares more about rendering perceptual details to obtain the target Y. For example, in text-to-image synthesis, $X \rightarrow Y'$ converts source text into discrete visual tokens that describe the semantic meanings of the target image, and $Y' \rightarrow Y$ renders the image details from visual tokens. How to design a learning mechanism to better disentangle the semantic meaning and perceptual details and let Y' focus on semantic meaning will be a good research opportunity.

- **RQ5**: How to disentangle semantic meaning and perceptual details to learn a semantic instead of detailed Y'?

Discrete vs. Continuous Y'. Both discrete and continuous Y' can be leveraged to bridge the mapping between X and Y. For example, using similar VQ-VAE to quantize speech waveforms, DelightfulTTS 2 (Liu et al. 2022) leverages continuous vectors as Y' while DiscreTalk (Hayashi and Watanabe 2020) leverages discrete tokens as Y'. Which kinds of representations (discrete vs continuous) are better choices for Y' is also an interesting point to investigate.

- **RQ6**: How to determine the discrete or continuous format of Y' for each data generation task?

How to Learn $X \rightarrow Y'$ and $Y' \rightarrow Y$

Regeneration learning decomposes a conditional data generation task $X \rightarrow Y$ into data conversion and data rendering processes. The data conversion process converts source data X into the target domain, which maintains the concrete semantics but does not necessarily contain fine-grained details. The data rendering process further renders the fine-grained details of the target data to achieve high-quality data generation. Roughly speaking, $X \rightarrow Y'$ undertakes more on the role of data conversion, while $Y' \rightarrow Y$ undertakes more on the role of data rendering.

How to design better training methods on $X \rightarrow Y'$ and $Y' \rightarrow Y$ would be important for the final performance of $X \rightarrow Y$ mapping in regeneration learning. Advanced generative models such as autoregressive models (Oord et al. 2016; Brown et al. 2020), VAEs (Kingma and Welling 2013), GANs (Goodfellow et al. 2014), normalizing flows (Dinh, Krueger, and Bengio 2014; Rezende and Mohamed 2015), and diffusion models (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020) can play an important role. Furthermore, considering $Y' \rightarrow Y$ mapping can usually be learned through large-scale self-supervised methods, we should leverage more unpaired data when learning $Y' \rightarrow Y$.

- **RQ7**: How to design better generative models to learn $X \rightarrow Y'$ and $Y' \rightarrow Y$ mapping?
- **RQ8**: How to leverage the assumption of semantic conversion in $X \rightarrow Y'$ and detail rendering in $Y' \rightarrow Y$ to design better methods?
- **RQ9**: How to leverage large-scale self-supervised learning for $Y' \rightarrow Y$ mapping?

How to Reduce Training-Inference Mismatch in $X \rightarrow Y' \rightarrow Y$

The model of $Y' \rightarrow Y$ is trained in a self-supervised way, where Y' is extracted from Y in training. However, Y' is predicted from X in inference, which causes the training-inference mismatch. How to reduce the mismatch is important to ensure the performance of this cascaded system ($X \rightarrow Y'$ and $Y' \rightarrow Y$). A straightforward way is to design an end-to-end optimization method between $X \rightarrow Y'$ and $Y' \rightarrow Y$, but still maintain Y' as an intermediate representation. For example, NaturalSpeech (Tan et al. 2022) leverages VAEs and normalizing flows with bidirectional prior/posterior optimization to achieve end-to-end learning. Can we design other methods to reduce the training-inference mismatch in regeneration learning?

- **RQ10**: How to reduce the training-inference mismatch in regeneration learning?

Conclusion

In this paper, we present a learning paradigm called regeneration learning for data generation tasks. Literally, it means generating the data two times: first generates an intermediate representation Y' from source data X, and then generates a target data Y from Y'. Metaphorically, it means generating Y' as intermediate representations to signify Y, in analogy to representation learning. Regeneration learning can be regarded as a counterpart of traditional representation learning: regeneration learning handles the abstraction (Y') of the target data Y for data generation while representation learning handles the abstraction (X') of source data X for data understanding, and both the processes of $Y' \rightarrow Y$ in regeneration learning and $X \rightarrow X'$ in representation learning can be learned in a self-supervised way (it is also a counterpart in literally: presentation→representation vs. generation→regeneration). We discuss the connections of regeneration learning to other methods, demonstrate a variety of data generation tasks that can benefit from regeneration learning, and further point out some research opportunities on regeneration learning. Regeneration learning can be a widely used paradigm for high-quality data generation and can provide valuable insights into developing data generation methods.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Benesty, J.; Sondhi, M. M.; Huang, Y.; et al. 2008. *Springer handbook of speech processing*, volume 1. Springer.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE TPAMI*, 35(8): 1798–1828.

Bertero, M.; Poggio, T. A.; and Torre, V. 1988. Ill-posed problems in early vision. *Proceedings of the IEEE*, 76(8): 869–889.

Blanz, V.; and Vetter, T. 1999. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 187–194.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Chan, W.; Jaitly, N.; Le, Q.; and Vinyals, O. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4960–4964. IEEE.

Chang, H.; Zhang, H.; Barber, J.; Maschinot, A.; Lezama, J.; Jiang, L.; Yang, M.-H.; Murphy, K.; Freeman, W. T.; Rubinstein, M.; et al. 2023. Muse: Text-To-Image Generation via Masked Generative Transformers. *arXiv preprint arXiv:2301.00704*.

Chen, L.; Maddox, R. K.; Duan, Z.; and Xu, C. 2019. Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7832–7841.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE): 2493–2537.

Cong, J.; Yang, S.; Xie, L.; and Su, D. 2021. Glow-WaveGAN: Learning Speech Representations from GAN-based Variational Auto-Encoder For High Fidelity Flow-based Speech Synthesis. *arXiv preprint arXiv:2106.10831*.

Dai, S.; Jin, Z.; Gomes, C.; and Dannenberg, R. B. 2021. Controllable deep melody generation via hierarchical music structure representation. *arXiv preprint arXiv:2109.00663*.

Défossez, A.; Copet, J.; Synnaeve, G.; and Adi, Y. 2022. High Fidelity Neural Audio Compression. *arXiv preprint arXiv:2210.13438*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Di, S.; Jiang, Z.; Liu, S.; Wang, Z.; Zhu, L.; He, Z.; Liu, H.; and Yan, S. 2021. Video Background Music Generation with Controllable Music Transformer. In *Proceedings of the 29th ACM International Conference on Multimedia*, 2037–2045.

Ding, M.; Yang, Z.; Hong, W.; Zheng, W.; Zhou, C.; Yin, D.; Lin, J.; Zou, X.; Shao, Z.; Yang, H.; et al. 2021. CogView: Mastering Text-to-Image Generation via Transformers. *arXiv preprint arXiv:2105.13290*.

Dinh, L.; Krueger, D.; and Bengio, Y. 2014. Nice: Nonlinear independent components estimation. *arXiv preprint arXiv:1410.8516*.

Dong, X.; Bao, J.; Zhang, T.; Chen, D.; Zhang, W.; Yuan, L.; Chen, D.; Wen, F.; and Yu, N. 2021. PeCo: Perceptual Codebook for BERT Pre-training of Vision Transformers. arXiv:2111.12710.

Esser, P.; Rombach, R.; and Ommer, B. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12873–12883.

Forsyth, D.; and Ponce, J. 2011. *Computer vision: A modern approach.* Prentice hall.

Girshick, R. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Graves, A. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Gray, R. 1984. Vector quantization. *IEEE Assp Magazine*, 1(2): 4–29.

Hassan, H.; Aue, A.; Chen, C.; Chowdhary, V.; Clark, J.; Federmann, C.; Huang, X.; Junczys-Dowmunt, M.; Lewis, W.; Li, M.; et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

Hayashi, T.; and Watanabe, S. 2020. Discretalk: Text-to-speech as a machine translation problem. *arXiv preprint arXiv:2005.05525*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hinton, G.; Deng, L.; Yu, D.; Dahl, G. E.; Mohamed, A.-r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T. N.; et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6): 82–97.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33: 6840–6851.

Ji, X.; Zhou, H.; Wang, K.; Wu, W.; Loy, C. C.; Cao, X.; and Xu, F. 2021. Audio-driven emotional video portraits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14080–14089.

Ju, Z.; Lu, P.; Tan, X.; Wang, R.; Zhang, C.; Wu, S.; Zhang, K.; Li, X.; Qin, T.; and Liu, T.-Y. 2021. TeleMelody: Lyric-to-Melody Generation with a Template-Based Two-Stage Method. *arXiv preprint arXiv:2109.09617*.

Kingma, D. P.; and Dhariwal, P. 2018. Glow: generative flow with invertible $1\times$ 1 convolutions. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 10236–10245.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kreuk, F.; Synnaeve, G.; Polyak, A.; Singer, U.; Défossez, A.; Copet, J.; Parikh, D.; Taigman, Y.; and Adi, Y. 2022. Audiogen: Textually guided audio generation. *arXiv preprint arXiv:2209.15352*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25: 1097–1105.

Lahiri, A.; Kwatra, V.; Frueh, C.; Lewis, J.; and Bregler, C. 2021. LipSync3D: Data-Efficient Learning of Personalized 3D Talking Faces from Video using Pose and Lighting Normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2755–2764.

Ling, J.; Tan, X.; Chen, L.; Li, R.; Zhang, Y.; Zhao, S.; and Song, L. 2022. StableFace: Analyzing and Improving Motion Stability for Talking Face Generation. *arXiv preprint arXiv:2208.13717*.

Liu, Y.; Xue, R.; He, L.; Tan, X.; and Zhao, S. 2022. DelightfulTTS 2: End-to-End Speech Synthesis with Adversarial Vector-Quantized Auto-Encoders. *arXiv preprint arXiv:2207.04646*.

Manning, C.; and Schutze, H. 1999. *Foundations of statistical natural language processing*. MIT press.

McKeown, K. 1992. *Text generation*. Cambridge University Press.

Oord, A. v. d.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; and Kavukcuoglu, K. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Preechakul, K.; Chatthee, N.; Wizadwongsa, S.; and Suwajanakorn, S. 2022. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10619–10629.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Rakhimov, R.; Volkhonskiy, D.; Artemov, A.; Zorin, D.; and Burnaev, E. 2020. Latent video transformer. *arXiv preprint arXiv:2006.10704*.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot text-to-image generation. In *ICML*, 8821–8831. PMLR.

Razavi, A.; van den Oord, A.; and Vinyals, O. 2019. Generating diverse high-fidelity images with vq-vae-2. In *Advances in neural information processing systems*, 14866–14876.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.

Ren, Y.; Ruan, Y.; Tan, X.; Qin, T.; Zhao, S.; Zhao, Z.; and Liu, T.-Y. 2019. Fastspeech: Fast, robust and controllable text to speech. *arXiv preprint arXiv:1905.09263*.

Rezende, D.; and Mohamed, S. 2015. Variational inference with normalizing flows. In *International Conference on Machine Learning*, 1530–1538. PMLR.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10684–10695.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E.; Ghasemipour, S. K. S.; Ayan, B. K.; Mahdavi, S. S.; Lopes, R. G.; et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*.

Shen, J.; Pang, R.; Weiss, R. J.; Schuster, M.; Jaitly, N.; Yang, Z.; Chen, Z.; Zhang, Y.; Wang, Y.; Skerrv-Ryan, R.; et al. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4779–4783. IEEE.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2256–2265. PMLR.

Song, L.; Liu, B.; Yin, G.; Dong, X.; Zhang, Y.; and Bai, J.-X. 2021. TACR-Net: Editing on Deep Video and Voice Portraits. In *Proceedings of the 29th ACM International Conference on Multimedia*, 478–486.

Sun, H.; Tan, X.; Gan, J.-W.; Liu, H.; Zhao, S.; Qin, T.; and Liu, T.-Y. 2019. Token-level ensemble distillation for grapheme-to-phoneme conversion. *arXiv preprint arXiv:1904.03446*.

Szeliski, R. 2010. *Computer vision: algorithms and applications*. Springer Science & Business Media.

Tan, X.; Chen, J.; Liu, H.; Cong, J.; Zhang, C.; Liu, Y.; Wang, X.; Leng, Y.; Yi, Y.; He, L.; et al. 2022. NaturalSpeech: End-to-End Text to Speech Synthesis with Human-Level Quality. *arXiv preprint arXiv:2205.04421*.

Tan, X.; Qin, T.; Soong, F.; and Liu, T.-Y. 2021. A survey on neural speech synthesis. *arXiv preprint arXiv:2106.15561*.

Thies, J.; Elgharib, M.; Tewari, A.; Theobalt, C.; and Nießner, M. 2020. Neural voice puppetry: Audio-driven facial reenactment. In *European Conference on Computer Vision*, 716–731. Springer.

van den Oord, A.; Vinyals, O.; and Kavukcuoglu, K. 2017. Neural discrete representation learning. In *Proceedings of*

the 31st International Conference on Neural Information Processing Systems, 6309–6318.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, Y.; Skerry-Ryan, R.; Stanton, D.; Wu, Y.; Weiss, R. J.; Jaitly, N.; Yang, Z.; Xiao, Y.; Chen, Z.; Bengio, S.; et al. 2017. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*.

Wu, J.; Liu, X.; Hu, X.; and Zhu, J. 2020. PopMNet: Generating structured pop music melodies using neural networks. *Artificial Intelligence*, 286: 103303.

Yan, W.; Zhang, Y.; Abbeel, P.; and Srinivas, A. 2021. VideoGPT: Video Generation using VQ-VAE and Transformers. *arXiv preprint arXiv:2104.10157*.

Yi, R.; Ye, Z.; Zhang, J.; Bao, H.; and Liu, Y.-J. 2020. Audio-driven talking face video generation with learning-based personalized head pose. *arXiv preprint arXiv:2002.10137*.

Yu, J.; Xu, Y.; Koh, J. Y.; Luong, T.; Baid, G.; Wang, Z.; Vasudevan, V.; Ku, A.; Yang, Y.; Ayan, B. K.; et al. 2022. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*.

Zeghidour, N.; Luebs, A.; Omran, A.; Skoglund, J.; and Tagliasacchi, M. 2021. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30: 495–507.

Zhang, Y.; and Wallace, B. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2223–2232.

Zou, Y.; Zou, P.; Zhao, Y.; Zhang, K.; Zhang, R.; and Wang, X. 2021. MELONS: generating melody with long-term structure using transformers and structure graph. *arXiv preprint arXiv:2110.05020*.