Voice2Series: Reprogramming Acoustic Models for Time Series Classification

Chao-Han Huck Yang [1] **Yun-Yun Tsai** [2] **Pin-Yu Chen** [3]

Abstract

Learning to classify time series with limited data is a practical yet challenging problem. Current methods are primarily based on hand-designed feature extraction rules or domain-specific data augmentation. Motivated by the advances in deep speech processing models and the fact that voice data are univariate temporal signals, in this paper we propose *Voice2Series* (V2S), a novel end-to-end approach that reprograms acoustic models for time series classification, through input transformation learning and output label mapping. Leveraging the representation learning power of a large-scale pre-trained speech processing model, on 30 different time series tasks we show that V2S either outperforms or is tied with state-of-the-art methods on 20 tasks, and improves their average accuracy by 1.84%. We further provide a theoretical justification of V2S by proving its population risk is upper bounded by the source risk and a Wasserstein distance accounting for feature alignment via reprogramming. Our results offer new and effective means to time series classification. Our code is available at https://github.com/huckiyang/Voice2Series-Reprogramming.

1. Introduction

Machine learning for time series data has rich applications in a variety of domains, ranging from medical diagnosis (e.g., physiological signals such as electrocardiogram (ECG) (Kampouraki et al., 2008)), finance/weather forecasting, to industrial measurements (e.g., sensors and Internet of Things (IoT)). It is worth noting that one common practical challenge that prevents time series learning tasks from using modern large-scale deep learning models is data scarcity.

[1]Georgia Institute of Technology [2]Columbia University [3]IBM Research. Correspondence to: Chao-Han Huck Yang <huckiyang@gatech.edu>, Pin-Yu Chen <pin-yu.chen@ibm.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

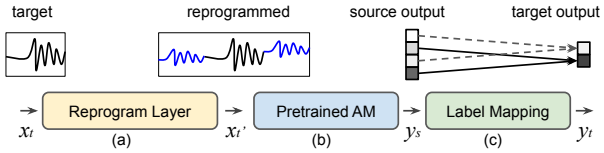


Figure 1. Schematic illustration of the proposed Voice2Series (V2S) framework: (a) trainable reprogram layer; (b) pre-trained acoustic model (AM); (c) source-target label mapping function.

While many efforts (Fawaz et al., 2018; Ye & Dai, 2018; Kashiparekh et al., 2019) have been made to advance transfer learning and model adaptation for time series classification, a principled approach is lacking and its performance may not be comparable to conventional statistical learning benchmarks (Langkvist et al., 2014).

To bridge this gap, we propose a novel approach, named **voice to series (V2S)**, for time series classification by *reprogramming* a pre-trained acoustic model (AM), such as a spoken-terms recognition model. Unlike general time series tasks, modern AMs are trained on massive human voice datasets and are considered as a mature technology widely deployed in intelligent electronic devices. The rationale of V2S lies in the fact that voice data can be viewed as univariate temporal signals, and therefore a well-trained AM is likely to be reprogrammed as a powerful feature extractor for solving time series classification tasks. Figure 1 shows a schematic illustration of the proposed V2S framework, including (a) a trainable reprogram layer, (b) a pre-trained AM, and (c) a specified label mapping function between source (human voice) and target (time series) labels.

Model reprogramming was firstly introduced in (Elsayed et al., 2019). The authors show that one can learn a universal input transformation function to reprogram a pre-trained ImageNet model (without changing the model weights) for solving MNIST/CIFAR-10 image classification and simple vision-based counting tasks with high accuracy. It can be viewed as an efficient approach for transfer learning with limited data, and it has achieved state-of-the-art (SOTA) results on biomedical image classification tasks (Tsai et al., 2020). However, despite the empirical success, little is known on how and why reprogramming can be successful.

Different from existing works, this paper aims to address the following three fundamental questions: (i) Can acous-

tic models be reprogrammed for time series classification? (ii) Can V2S outperform SOTA time-series classification results? (iii) Is there any theoretical justification on why reprogramming works?

Our main contributions in this paper provide affirmative answers to the aforementioned fundamental questions, which are summarized as follows.

1. We propose V2S, a novel and unified approach to reprogram large-scale pre-trained acoustic models for different time series classification tasks. To the best of our knowledge, V2S is the first framework that enables reprogramming for time series tasks.

2. Tested on a standard UCR time series classification benchmark (Dau et al., 2019), V2S either outperforms or is tied with the best reported results on 20 out of 30 datasets and improves their average accuracy by 1.84%, suggesting that V2S is a principled and effective approach for time series classification.

3. In Section 4, we develop a theoretical risk analysis to characterize the performance of reprogramming on the target task via source risk and representation alignment loss. In Section 5, we also show how our theoretical results can be used to assess the performance of reprogramming. Moreover, we provide interpretation on V2S through auditory neural saliency map and embedding visualization.

2. Related Works

2.1. Time Series Classification

Learning to classify time series is a standard research topic in machine learning and signal processing. A major research branch uses designed features followed by conventional classifiers, such as digital filter design in together with support vector machine (SVM) (Kampouraki et al., 2008), decision-trees (Geurts, 2001), or kernel based methods (Zhang et al., 2010; Lines et al., 2012). Recently, deep learning models have been utilized in time series (Fawaz et al., 2019) and demonstrated improved performance. The methods range from a completely end-to-end classifier (Zhang et al., 2010; Wang et al., 2017) to a mixture model (Hong et al., 2019) that combines feature engineering and deep learning. Notably, feature engineering methods (Wang et al., 2017) can still attain SOTA results on some time series classification tasks, especially when the number of training data is small.

2.2. Model Reprogramming

Although in the original paper (Elsayed et al., 2019) model reprogramming was phrased as "adversarial" reprogramming, it is not limited to the adversarial setting, nor does it involve any adversarial training. Elsayed et al. (2019) showed that the pre-trained ImageNet models can be reprogrammed for classifying other image datasets and for solving vision-based counting tasks. Tsai et al. (2020) demonstrated the advantage of reprogramming on label-limited data such as biomedical image classification, and used zeroth order optimization (Liu et al., 2020) to enable reprogramming of black-box machine learning systems.

Beyond image data, model reprogramming has been used in natural language processing (NLP) (Neekhara et al., 2019; Hambardzumyan et al., 2020), such as machine translation and sentiment classification. Vinod et al. (2020) further showed that NLP models can be reprogrammed for molecule learning tasks in biochemistry.

For reprogramming with time series, it remains unclear what source domain and pre-trained models to be used. Current research works on reprogramming do not extend to time series because image and text data have fundamentally different feature characteristics than time series. One of our major contributions is the proposal of reprogramming pre-trained acoustic models on abundant human voice data for time series classification. To our best knowledge, our V2S is the first framework on reprogramming for time series.

2.3. Deep Acoustic Modeling

Recent deep learning models have shown impressive results on predicting label(s) from acoustic information. The central idea is to use a large number of spectral features (e.g., Mel-spectrogram or log-power spectrum) as training inputs to capture the important features. Some of the latent features learned from AM are interpretable based on physiological auditory experiments (e.g., cortex responses (Kaya & Elhilali, 2017)) or neural saliency methods. Several efforts have been put into the design of a large and deep neural network to extract features from human voice datasets. Among them, residual neural network (ResNet) (He et al., 2016; Saon et al., 2017) and VGGish (Hershey et al., 2017) models are popular backbones for AM tasks, such as spoken-term recognition (Yang et al., 2021a; de Andrade et al., 2018) and speech enhancement (Xu et al., 2014; Yang et al., 2020). It is worth noting that standard transfer learning via parameter finetuning is not ideal for time series tasks with limited data, as acoustic data and time-series data are often quite diverse in scale. Our V2S addresses this issue via learning an input transformation function while fixing the pre-trained AM.

3. Voice2Series (V2S)

3.1. Mathematical Notation

Table 1 summarizes the major mathematical notations used in this paper for V2S reprogramming. Throughout this paper, we will denote a K-way acoustic classification model

Table 1. Mathematical notation for reprogramming

Symbol	Meaning
$\mathcal{S}$ / $\mathcal{T}$	source/target domain
$\mathcal{X}_\mathcal{S}$ / $\mathcal{X}_\mathcal{T}$	the space of source/target data samples
$\mathcal{Y}_\mathcal{S}$ / $\mathcal{Y}_\mathcal{T}$	the space of source/target data labels
$\mathcal{D}_\mathcal{S} \subseteq \mathcal{X}_\mathcal{S} \times \mathcal{Y}_\mathcal{S}$ / $\mathcal{D}_\mathcal{T} \subseteq \mathcal{X}_\mathcal{T} \times \mathcal{Y}_\mathcal{T}$	source/target data distribution
$(x, y) \sim \mathcal{D}$	data sample x and one-hot coded label y drawn from $\mathcal{D}$
K	number of source labels
$f_\mathcal{S} : \mathbb{R}^d \mapsto [0, 1]^K$	pre-trained K-way source classification model
$\eta : \mathbb{R}^K \mapsto [0, 1]^K$	softmax function in neural network, and $\sum_{k=1}^K [\eta(\cdot)]_k = 1$
$z(\cdot) \in \mathbb{R}^K$	logit (pre-softmax) representation, and $f(x) = \eta(z(x))$
$\ell(x, y) \triangleq \|f(x) - y\|_2$	risk function of (x, y) based on classifier f
$\mathbb{E}_\mathcal{D}[\ell(x, y)] \triangleq \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(x, y)] = \mathbb{E}_\mathcal{D}\|f(x) - y\|_2$	population risk based on classifier f
δ, θ	additive input transformation on target data, parameterized by θ

pre-trained on voice data as a *source* model, and use the term *target* data to denote the univariate time-series data to be reprogrammed. The notation P is reserved for denoting a probability function. The remaining notations will be introduced when applicable.

3.2. V2S Reprogramming on Data Inputs

Here we formulate the problem of V2S reprogramming on data inputs. Let $x_t \in \mathcal{X}_\mathcal{T} \subseteq \mathbb{R}^{d_\mathcal{T}}$ denote a univariate time series input from the target domain with $d_\mathcal{T}$ temporal features. Our V2S aims to find a trainable input transformation function $\mathcal{H}$ that is universal to all target data inputs, which serves the purpose of reprogramming x_t into the source data space $\mathcal{X}_\mathcal{S} \subseteq \mathbb{R}^{d_\mathcal{S}}$, where $d_\mathcal{T} < d_\mathcal{S}$. Specifically, the reprogrammed sample x'_t is formulated as:

$$x'_t = \mathcal{H}(x_t; \theta) := \text{Pad}(x_t) + \underbrace{M \odot \theta}_{\triangleq\ \delta} \tag{1}$$

where $\text{Pad}(x_t)$ is a zero padding function that outputs a zero-padded time series of dimension $d_\mathcal{S}$. The location of the segment x_t to be placed in x'_t is a design parameter and we defer the discussion to Section 5.2. The term $M \in \{0, 1\}^{d_\mathcal{S}}$ is a binary mask that indicates the location of x_t in its zero-padded input $\text{Pad}(x_t)$, where the i-th entry of M is 0 if x_t is present (indicating the entry is non-reprogrammable), and it is 1 otherwise (indicating the entry is not occupied and thus reprogrammable). The $\odot$ operator denotes element-wise product. Finally, $\theta \in \mathbb{R}^{d_\mathcal{S}}$ is a set of trainable parameters for aligning source and target domain data distributions. One can consider a more complex function $W(\theta)$ in our reprogramming function. But in practice we do not observe notable gains when compared to the simple function θ. In what follows, we will use the term $\delta \triangleq M \odot \theta$ to denote the trainable additive input transformation for V2S reprogramming. Moreover, for ease of representation we will omit the padding notation and simply use $x_t + \delta$ to denote the reprogrammed target data, by treating the "$+$" operation as a zero-padded broadcasting function.

3.3. V2S Reprogramming on Acoustic Models (AMs)

We select a pre-trained deep acoustic classification model as the source model ($f_\mathcal{S}$) for model reprogramming. We assume the source model has softmax as the final layer and outputs nonnegative confidence score (prediction probability) for each source label. With the transformed data inputs $\mathcal{H}(x_t; \theta)$ described in (1), one can obtain the class prediction of the source model $f_\mathcal{S}$ on an reprogrammed target data sample x_t, denoted by

$$P(y_s | f_\mathcal{S}(\mathcal{H}(x_t; \theta))), \text{ for all } y_s \in \mathcal{Y}_\mathcal{S} \tag{2}$$

Next, as illustrated in Figure 1, we assign a (many-to-one) label mapping function h to map source labels to target labels. For a target label $y_t \in \mathcal{Y}_\mathcal{T}$, its class prediction will be the averaged class predictions over the set of source labels assigned to it. We use the term $P(h(\mathcal{Y}_\mathcal{S}) | f_\mathcal{S}(\mathcal{H}(x_t; \theta)))$ to denote the prediction probability of the target task on the associated ground-truth target label $y_t = h(\mathcal{Y}_\mathcal{S})$. Finally, we learn the optimal parameters θ^* for data input reprogramming by optimizing the following objective:

$$\theta^* = \arg\min_\theta \underbrace{-\log P(h(\mathcal{Y}_\mathcal{S}) | f_\mathcal{S}(\mathcal{H}(x_t; \theta))}_{\text{V2S loss } \triangleq\ L}; \tag{3}$$
$$\text{where} \quad h(\mathcal{Y}_\mathcal{S}) = y_t$$

The optimization will be implemented by minimizing the empirical loss (V2S loss L) evaluated on all target-domain training data pairs $\{x_t, y_t\}$ for solving θ^*.

In practice, we find that many-to-one label mapping can improve the reprogramming accuracy when compared to one-to-one label mapping, similar to the findings in (Tsai et al., 2020). Below we make a concrete example on how

many-to-one label mapping is used for V2S reprogramming. Consider the case of reprogramming spoken-term AM for ECG classification. One can choose to map multiple (but non-overlapping) classes from the source task (e.g., 'yes', 'no', 'up', 'down' in AM classes) to every class from the target task (e.g., 'Normal' or 'Ischemia' in ECG classes), leading to a specified mapping function h. Let $\mathcal{B} \subset \mathcal{Y}_\mathcal{S}$ denote the set of source labels mapping to the target label $y_t \in \mathcal{Y}_\mathcal{T}$. Then, the class prediction of y_t based on V2S reprogramming is the aggregated prediction over the assigned source labels, which is defined as

$$P(y_t|f_\mathcal{S}(\mathcal{H}(x_t;\theta)) = \frac{1}{|\mathcal{B}|}\sum_{y_s\in\mathcal{B}} P(y_s|f_\mathcal{S}(\mathcal{H}(x_t;\theta)) \quad (4)$$

where $|\mathcal{B}|$ denotes the number of labels in $\mathcal{B}$. In our implementation we use random (but non-overlapping) many-to-one mapping between source and target labels. Each target label is assigned with the same number of source labels.

3.4. V2S Algorithm

Algorithm 1 summarizes the training procedure of our proposed V2S reprogramming algorithm. The algorithm uses the ADAM optimizer (Kingma & Ba, 2015) to find the optimal reprogramming parameters θ^* that minimize the V2S loss L as defined in (3), which is evaluated over all target-domain training data. In our implementation of Algorithm 1 we use stochastic optimization with minibatches.

Algorithm 1 Voice to Series (V2S) Reprogramming

Inputs: Pre-trained acoustic model $f_\mathcal{S}$, V2S loss L in (3), target domain training data $\{x_t^{(i)}, y_t^{(i)}\}_{i=1}^n$, mask function M, multi-label mapping function $h(\cdot)$, maximum number of iterations T, initial learning rate α
Output: Optimal reprogramming parameters θ^*
Initialize θ randomly; set $t = 0$
#Generate reprogrammed data input
$\mathcal{H}(x_t^{(i)};\theta) = \text{Pad}(x_t^{(i)}) + M \odot \theta, \forall\, i = \{1, 2, \ldots, n\}$
#Compute V2S loss L from equation (3)
$L(\theta) = -\frac{1}{n}\sum_{i=1}^n \log P(y_t^{(i)}|f_\mathcal{S}(\mathcal{H}(x_t^{(i)});\theta))$
#Solve reprogramming parameters
Use ADAM optimizer to solve for θ^* based on $L(\theta)$

4. Population Risk via Reprogramming

To provide theoretical justification on the effectiveness of V2S, in what follows we establish a formal population risk analysis and prove that based on V2S, the population risk of the target task is upper bounded by the sum of the source population risk and the Wasserstein-1 distance between the logit representations of the source data and the reprogrammed target data. Our analysis matches the intuition that a high-accuracy (low population risk) source model with a better source-target data alignment (small Wasserstein-1 distance) should exhibit better reprogramming performance. In Section 5.4, we show that our derived population risk bound can be used to assess the reprogramming performance of V2S for different source models and target tasks. We also note that our theoretical analysis is not limited to V2S reprogramming. It applies to generic classification tasks.

Using the mathematical notation summarized in Table 1, the source model is a pre-trained K-way neural network classifier $f_\mathcal{S}(\cdot) = \eta(z_\mathcal{S}(\cdot))$ with a softmax layer $\eta(\cdot)$ as the final model output. We omit the notation of the model parameters in our analysis because reprogramming does not change the pre-trained model parameters. The notation (x, y) is used to describe a data sample x and its one-hot coded label y. We will use the subscript s/t to denote source/target data when applicable. For the purpose of analysis, given a neural network classifier f, we consider the root mean squared error (RMSE) denoted by $\|f(x) - y\|_2$.

To put forth our analysis, we make the following assumptions based on the framework of reprogramming:

1. The source risk is $\epsilon_\mathcal{S}$, that is, $\mathbb{E}_{\mathcal{D}_\mathcal{S}}[\ell(x_s, y_s)] = \epsilon_\mathcal{S}$.

2. The source-target label space has a specified surjective one-to-one label mapping function h_t for every target label t, such that $\forall y_t \in \mathcal{Y}_\mathcal{T}$, $y_t = h_t(\mathcal{Y}_\mathcal{S}) \triangleq y_s \in \mathcal{Y}_\mathcal{S}$, and $h_t \neq h_{t'}$ if $t \neq t'$.

3. Based on reprogramming, the target loss function $\ell_\mathcal{T}$ with an additive input transformation function δ can be represented as $\ell_\mathcal{T}(x_t + \delta, y_t) \overset{(a)}{=} \ell_\mathcal{T}(x_t + \delta, y_s) \overset{(b)}{=} \ell_\mathcal{S}(x_t + \delta, y_s)$, where (a) is induced by label mapping (Assumption 2) and (b) is induced by reprogramming the source loss with target data.

4. The learned input transformation function for reprogramming is denoted by $\delta^* \triangleq \arg\min_\delta \mathbb{E}_{\mathcal{D}_\mathcal{T}}[\ell_\mathcal{S}(x_t + \delta, y_s)]$, which is the minimizer of the target population risk with the reprogramming loss objective.

5. Domain-independent drawing of source and target data: Let $\Phi_\mathcal{S}(\cdot)$ and $\Phi_\mathcal{T}(\cdot)$ denote the probability density function of source data and target data distributions over $\mathcal{X}_\mathcal{S}$ and $\mathcal{X}_\mathcal{T}$, respectively. The joint probability density function is the product of their marginals, i.e., $\Phi_{\mathcal{S},\mathcal{T}}(x_s, x_t) = \Phi_\mathcal{S}(x_s) \cdot \Phi_\mathcal{T}(x_t)$.

For a given neural network classifier, the following lemma associates the expected RMSE of model predictions on two different domains with the Wasserstein-1 distance between their corresponding probability measures on the logit representations, which will play a key role in characterizing the population risk for reprogramming. Wasserstein distance

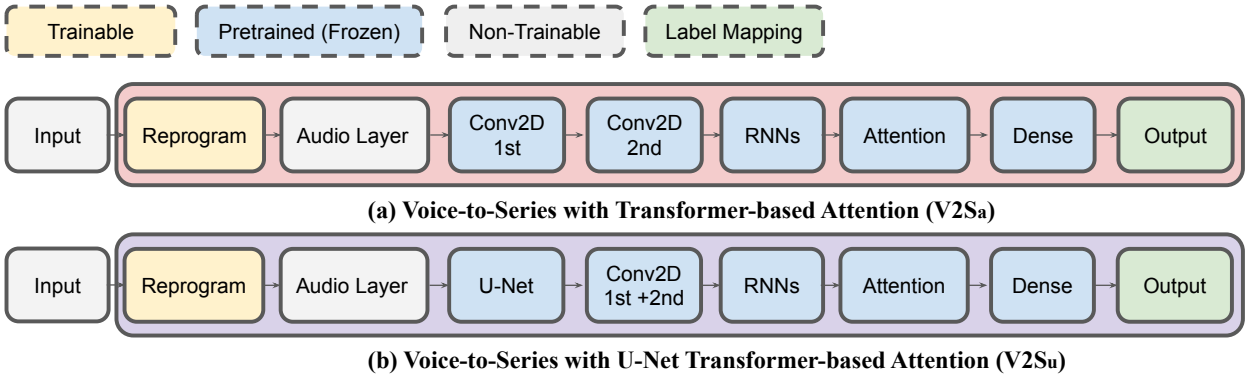


Figure 2. V2S architectures: (a) V2S$_a$ (de Andrade et al., 2018) and (b) V2S$_u$ (Yang et al., 2021a).

is a statistical distance between two probability measures μ and μ', and it has been widely used for studying optimal transport problems (Peyré & Cuturi, 2018). Specifically, for any $p \geq 1$, the Wasserstein-p distance is defined as

$$\mathcal{W}_p(\mu, \mu') = \left(\inf_{\pi \in \Pi(\mu, \mu')} \int \|x - x'\|^p d\pi(x, x') \right)^{1/p},$$

where $\Pi(\mu, \mu')$ denotes all joint distributions π that have marginals μ and μ'.

Lemma 1: Given a K-way neural network classifier $f(\cdot) = \eta(z(\cdot))$. Let μ_z and μ'_z be the probability measures of the logit representations $\{z(x)\}$ and $\{z(x')\}$ from two data domains $\mathcal{D}$ and $\mathcal{D}'$, where $x \sim \mathcal{D}$ and $x' \sim \mathcal{D}'$. Assume independent draws for x and x', i.e., $\Phi_{\mathcal{D},\mathcal{D}'}(x, x') = \Phi_{\mathcal{D}}(x) \cdot \Phi_{\mathcal{D}'}(x')$. Then

$$\mathbb{E}_{x \sim \mathcal{D},\ x' \sim \mathcal{D}'} \|f(x) - f(x')\|_2 \leq 2\sqrt{K} \cdot \mathcal{W}_1(\mu_z, \mu'_z),$$

where $\mathcal{W}_1(\mu_z, \mu'_z)$ is the Wasserstein-1 distance between μ_z and μ'_z.

Proof: Please see Appendix A.

With Lemma 1, we now state the main theorem regarding an upper bound on population risk for reprogramming.

Theorem 1: Let δ^* denote the learned additive input transformation for reprogramming (Assumption 4). The population risk for the target task via reprogramming a K-way source neural network classifier $f_{\mathcal{S}}(\cdot) = \eta(z_{\mathcal{S}}(\cdot))$, denoted by $\mathbb{E}_{\mathcal{D}_{\mathcal{T}}}[\ell_{\mathcal{T}}(x_t + \delta^*, y_t)]$, is upper bounded by

$$\mathbb{E}_{\mathcal{D}_{\mathcal{T}}}[\ell_{\mathcal{T}}(x_t + \delta^*, y_t)] \leq \underbrace{\epsilon_{\mathcal{S}}}_{\text{source risk}} + 2\sqrt{K} \cdot \underbrace{\mathcal{W}_1(\mu(z_{\mathcal{S}}(x_t + \delta^*)), \mu(z_{\mathcal{S}}(x_s)))_{x_t \sim \mathcal{D}_{\mathcal{T}},\ x_s \sim \mathcal{D}_{\mathcal{S}}}}_{\text{representation alignment loss via reprogramming}}$$

Proof: Please see Appendix B.

Theorem 1 shows that the target population risk via reprogramming is upper bounded by the summation of two terms: (i) the source population risk $\epsilon_{\mathcal{S}}$, and (ii) the representation alignment loss in the logit layer between the source data $z_{\mathcal{S}}(x_s)$ and the reprogrammed target data $z_{\mathcal{S}}(x_t + \delta^*)$ based on the same source neural network classifier $f_{\mathcal{S}}(\cdot) = \eta(z_{\mathcal{S}}(\cdot))$, measured by their Wasserstein-1 distance. The results suggest that reprogramming can attain better performance (lower risk) when the source model has a lower source loss and a smaller representation alignment loss.

In the extreme case, if the source and target representations can be fully aligned, the Wasserstein-1 distance will become 0 and thus the target task via reprogramming can perform as well as the source task. On the other hand, if the representation alignment loss is large, then it may dominate the source risk and hinder the performance on the target task. In the next section we will investigate how the representation alignment loss can inform the reprogramming performance for V2S. We would also like to make a final remark that our risk analysis can be extended beyond the additive input transformation setting, by considering a more complex function input transformation function $g(x_t)$ (e.g., an affine transformation). However, in practice, we observe little gain of doing so in V2S and therefore focus on the additive input transformation setting.

5. Performance Evaluation

5.1. Acoustic Models (AMs) and Source Datasets

We start by introducing the source models and the source datasets they trained on. The pre-trained source models will be used in our V2S experiments.

Limited-vocabulary Voice Commends Dataset: To create a large-scale (∼100k training samples) pre-trained acoustic model for our experiments, we select the Google Speech Commands V2 (Warden, 2018) (denoted as GSCv2) dataset, which contains 105,829 utterances of 35 words from 2,618 recorded speakers with a sampling rate of 16 kHz. We also provide some discussion on models trained on other

acoustic datasets (e.g., AudioSet (Gemmeke et al., 2017)) in Appendix D. In general, using the same network architecture we find that AMs trained on the voice commands dataset show better V2S performance than other datasets, which could be attributed to its similarity to short-length (one-second or less) time series data.

Transformer-based AMs: For training the source model, we use a popular transformer based single-head self-attention architecture (de Andrade et al., 2018) for V2S reprogramming, denoted as V2S$_a$ (Figure 2 (a)). We also train a similar architecture with U-Net (Long et al., 2015), denoted as V2S$_u$ (Figure 2 (b)), which is designed to enhance feature extraction in acoustic tasks (Yang et al., 2021a). Both pretrained V2S$_a$ and V2S$_u$ models have a comparable number ($\sim$0.2M/0.3M) of model parameters and test accuracy (96.90%/96.92%). Their data input dimension is $d \sim$16k and thus the reprogramming function θ has $\sim$16k trainable parameters. The details of the transformer-based models and comparisons to other popular neural network architectures are given in Appendix C.

5.2. V2S Implementation and Baseline

V2S Implementation: We use Tensorflow (Abadi et al., 2016) (v2.2) to implement our V2S framework following Algorithm 1. To enable end-to-end V2S training, we use the Kapre toolkit (Choi et al., 2017) to incorporate an on-GPU audio preprocessing layer, as shown in Figure 2. For the V2S parameters in Algorithm 1, we use $\alpha = 0.05$ and a mini-batch size of 32 with $T = 100$ training epochs. We use maximal many-to-one random label mapping, which assigns $\lfloor \frac{|\mathcal{Y}_\mathcal{S}|}{|\mathcal{Y}_\mathcal{T}|} \rfloor$ non-overlapping source labels to every target label, where $|\mathcal{Y}|$ is the size of the label set $\mathcal{Y}$ and $\lfloor z \rfloor$ is the floor function that gives the largest integer not exceeding z. To stabilize the training process, we add weight decay as a regularization term to the V2S loss and set the regularization coefficient to be 0.04. Our V2S implementation is open-source and available at `https://github.com/huckiyang/Voice2Series-Reprogramming`.

For *model tuning*, we use dropout during training on the reprogramming parameters θ. Moreover, during input reprogramming we also replicate the target signal x_t into m segments and place them starting from the beginning of the reprogrammed input with an identical interval (see Figure 4 (a) as an example with $m = 3$). For each task, we report the best result of V2S among a set of hyperparmeters with dropout rate $\in \{0, 0.1, 0.2, 0.3, 0.4\}$ and the number of target signal replication $m \in \{1, 2, \ldots, 10\}$. We use 10-fold splitting on training data to select the best performed model based on the validation loss and report the accuracy on test data with an average of 10 runs, which follows a similar experimental setting used in (Cabello et al., 2020).

Transfer Learning Baseline (TF$_a$): To demonstrate the effectivness of reprogramming, we also provide a transfer learning baseline using the same V2S$_a$ pre-trained model. Different from V2S, this baseline (named TF$_a$) does not use input reprogramming but instead allows fine-tuning the pre-trained model parameters using the zero-padded target data. An additional dense layer for task-dependent classification is also included for training.

5.3. UCR Time Series Classification Benchmark

UCR Archive (Dau et al., 2019) is a prominent benchmark that contains a large collection of time series classification datasets with default training and testing data splitting. The current state-of-the-art (SOTA) results on test accuracy are obtained from the following methods: (i) deep neural networks including fully convolutional networks (FCN) and deep residual neural networks as reported in (Wang et al., 2017); (ii) bag-of-features framework (Schäfer, 2015; Cabello et al., 2020); (iii) ensemble-based framework (Hills et al., 2014; Bagnall et al., 2015; Lines et al., 2018); (iv) time warping framework (Ratanamahatana & Keogh, 2005).

To ensure each target label is at least assigned with 3 unique source labels, we select 30 time series datasets in UCR Archive with the number of target labels ≤ 10 for our V2S experiments. We note that for each dataset, the algorithm that achieves the current SOTA result can vary, and therefore the comparison can be unfair to V2S. Nonetheless, the results still provide new sights on *how many datasets can V2S outperform and obtain the new SOTA*, rather than which method is best for time series classification?

In addition to comparing the standard test accuracy of each dataset as well as the mean and median accuracy over all datasets, we also report the mean per-class error (MPCE) proposed in (Wang et al., 2017), which is a single metric for performance evaluation over multiple datasets. MPCE is the sum of per-class error (PCE) over J datasets, definded as $\text{MPCE} = \sum_{j \in [J]} \text{PCE}_j = \frac{e_j}{c_j}$, which comprises of the error rate (e_j) and the number of classes (c_j) for each dataset.

Reprogramming Performance: Table 2 summarizes the performance of each method on 30 datasets. Notably, our reprogrammed V2S$_a$ model attains either better or equivalent results on 20 out of 30 time series datasets, suggesting that V2S as a single method is a competitive and promising approach for time series classification. The transfer learning baseline TF$_a$ has poor performance, which can be attributed to limited training data. V2S$_a$ has higher mean/median accuracy (accuracy increases by 1.84/2.63%) and lower MPCE (relative error decreases by about 2.87%) than that of SOTA results, demonstrating the effectiveness of V2S. For most datasets, V2S$_a$ has better performance than V2S$_u$, which can be explained by Theorem 1 through a lower empirical target risk upper bound (see Section 5.4).

Table 2. Performance comparison of test accuracy (%) on 30 UCR time series classification datasets (Dau et al., 2019). Our proposed $V2S_a$ outperforms or ties with the current SOTA results (discussed in Section 5.3) on 20 out of 30 datasets.

Dataset	Type	Input size	Train. Data	Class	SOTA	$V2S_a$	$V2S_u$	TF_a
Coffee	SPECTRO	286	28	2	**100**	**100**	**100**	53.57
DistalPhalanxTW	IMAGE	80	400	6	**79.28**	79.14	75.34	70.21
ECG 200	ECG	96	100	2	90.9	**100**	**100**	**100**
ECG 5000	ECG	140	500	5	**94.62**	93.96	93.11	58.37
Earthquakes	SENSOR	512	322	2	76.91	**78.42**	76.45	74.82
FordA	SENSOR	500	2500	2	96.44	**100**	**100**	**100**
FordB	SENSOR	500	3636	2	92.86	**100**	**100**	**100**
GunPoint	MOTION	150	50	2	**100**	96.67	93.33	49.33
HAM	SPECTROM	431	109	2	**83.6**	78.1	71.43	51.42
HandOutlines	IMAGE	2709	1000	2	**93.24**	**93.24**	91.08	64.05
Haptics	MOTION	1092	155	5	51.95	**52.27**	50.32	21.75
Herring	IMAGE	512	64	2	**68.75**	**68.75**	64.06	59.37
ItalyPowerDemand	SENSOR	24	67	2	97.06	**97.08**	96.31	97
Lightning2	SENSOR	637	60	2	86.89	**100**	**100**	**100**
MiddlePhalanxOutlineCorrect	IMAGE	80	600	2	72.23	**83.51**	81.79	57.04
MiddlePhalanxTW	IMAGE	80	399	6	58.69	**65.58**	63.64	27.27
Plane	SENSOR	144	105	7	**100**	**100**	**100**	9.52
ProximalPhalanxOutlineAgeGroup	IMAGE	80	400	3	88.09	**88.78**	87.8	48.78
ProximalPhalanxOutlineCorrect	IMAGE	80	600	2	**92.1**	91.07	90.03	68.38
ProximalPhalanxTW	IMAGE	80	400	6	81.86	**84.88**	83.41	35.12
SmallKitchenAppliances	DEVICE	720	375	3	**85.33**	83.47	74.93	33.33
SonyAIBORobotSurface	SENSOR	70	20	2	**96.02**	**96.02**	91.71	34.23
Strawberry	SPECTRO	235	613	2	**98.1**	97.57	91.89	64.32
SyntheticControl	SIMULATED	60	300	6	**100**	98	99	49.33
Trace	SENSOR	271	100	4	**100**	**100**	**100**	18.99
TwoLeadECG	ECG	82	23	2	**100**	96.66	97.81	49.95
Wafer	SENSOR	152	1000	2	99.98	**100**	**100**	100
WormsTwoClass	MOTION	900	181	2	83.12	**98.7**	90.91	57.14
Worms	MOTION	900	181	5	80.17	**83.12**	80.34	42.85
Wine	SPECTRO	234	57	2	**92.61**	90.74	90.74	50
Mean accuracy (↑)	-	-	-	-	88.02	**89.86**	87.92	56.97
Median accuracy (↑)	-	-	-	-	92.36	**94.99**	91.40	53.57
MPCE (mean per class error) (↓)	-	-	-	-	2.09	**2.01**	2.10	48.34

5.4. Representation Alignment Loss

According to Theorem 1, the target risk is upper bounded by the sum of a fixed source risk and a representation alignment loss between the source and reprogrammed target data. The latter is measured by the Wasserstein-1 distance of their logit representations. We use the following experiments to empirically verify the representation alignment loss during V2S training, and motivate its use for reprogramming performance assessment. Specifically, for computational efficiency we use the sliced Wasserstein-2 distance (SWD) (Kolouri et al., 2018) to approximate the Wasserstein-1 distance in Theorem 1. SWD uses one-dimensional (1D) random projection (we use 1,000 runs) to compute the sliced Wasserstein-2 distance by invoking 1D-optimal transport (OT), which possesses computational efficiency when compared to higher-dimensional OT problems (Peyré & Cuturi, 2018). Moreover, the Wasserstein-1 distance is upper bounded by the Wasserstein-2 distance (Peyré & Cuturi, 2018), and therefore the SWD will serve as a good approximation of the exact representation alignment loss.

Wasserstein Distance during Training: Using the DistalPhalanxTW (Davis, 2013) dataset and $V2S_a$ in Table 2, Figure 3 shows the validation (test) accuracy, validation (test) loss, and SWD during V2S training. One can observe a similar trend between test loss and SWD, suggesting that V2S indeed learns to reprogram the target data representations by gradually making them closer to the source data distribution, as indicated by Theorem 1.

Model Selection: Based on Theorem 1, one can leverage our derived risk bound for V2S model selection. Comparing $V2S_a$ and $V2S_u$, Table 3 shows the validation loss of the source task (GSCv2 voice dataset (Warden, 2018)) and the mean/median SWD over all 30 training sets of the target tasks in Table 2. We find that $V2S_a$ indeed has a lower sum

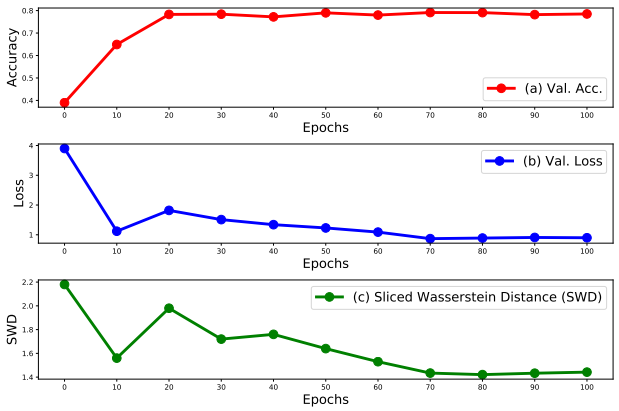


Figure 3. Training-time reprogramming analysis using V2S$_a$ and DistalPhalanxTW dataset (Davis, 2013). All values are averaged over the test set. The rows are (a) validation (test) accuracy, (b) validation loss, and (c) sliced Wasserstein distance (SWD) (Kolouri et al., 2018).

Table 3. Validation loss (Loss$_{\mathcal{S}}$) of the source task (GSCv2 voice dataset (Warden, 2018)) and mean/median Sliced Wasserstein Distance (SWD) of all training sets in Table 2.

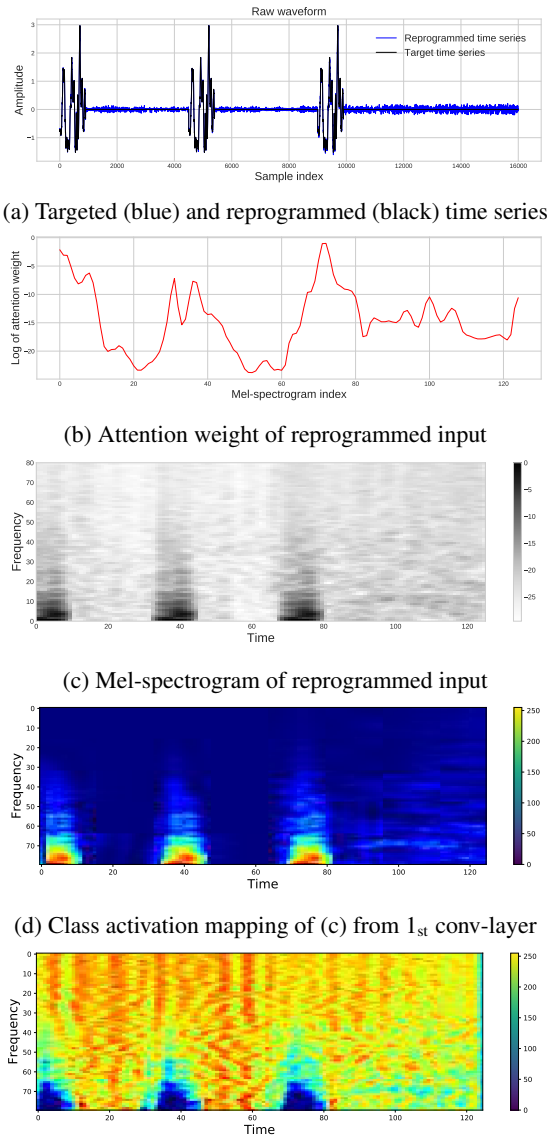
Model	Loss$_{\mathcal{S}}$ (↓)	Mean SWD (↓)	Median SWD (↓)
V2S$_a$	**0.1709**	**1.829**	**1.943**
V2S$_u$	0.1734	1.873	1.977

of the source loss and SWD than V2S$_u$, which explains its improved performance in Table 2.

5.5. Additional Analysis on V2S Interpretation

To gain further insights on V2S, we study its acoustic saliency map and embedding visualization.

Attention and Class Activation Mapping: To interpret the prediction made by V2S, we provide neural saliency analysis over the spectrogram of the reprogrammed features by class activation mapping (CAM) (Zhou et al., 2016) using the Worms dataset (Bagnall et al., 2015) and V2S$_a$. Activation and attention mapping methods (Wu & Lee, 2019) have been used in auditory analysis (Fritz et al., 2007) and investigated on its relationship between audio signal and brain cortex activation by neural physiology studies (Kaya & Elhilali, 2017; Veale et al., 2017). Interestingly, as shown in Figure 4, the corresponding attention head (b) of pre-trained AM (non-trainable during V2S process) could still recognize the original temporal patterns from the reprogrammed input signal in (a). We also show the Mel-spectrogram of reprogrammed input in (c), indicating the activated spatial-temporal acoustic features corresponding to weighted output prediction. Furthermore, in reference to the V2S architecture introduced in Figure 2, we select the first and the second convolution layer for CAM visualization in (d) and (e).



(a) Targeted (blue) and reprogrammed (black) time series

(b) Attention weight of reprogrammed input

(c) Mel-spectrogram of reprogrammed input

(d) Class activation mapping of (c) from 1$_{st}$ conv-layer

(e) Class activation mapping of (c) from 2$_{nd}$ conv-layer

Figure 4. Visualization of a data sample in the Worms dataset (Bagnall et al., 2015) using V2S$_a$. The rows are (a) the original target time series and its reprogrammed pattern as illustrated in Figure 1, (b) the associated attention-head predicted by V2S$_a$, (c) Mel-spectrogram of the reprogrammed input, and (d)/(e) its neural saliency maps via class activation mapping (Zhou et al., 2016) from the first/second convolution layer.

From the analysis, we observe different functions of these two layers. The first convolution layer tends to focus on the target signal segments themselves as well as their low-frequency acoustic features of the reprogrammed input's Mel-spectrogram in (c), whereas the second convolution layer tends to put more focus on the high-frequency components in the reprogrammed input.

Embedding Visualization: We use t-distributed stochas-

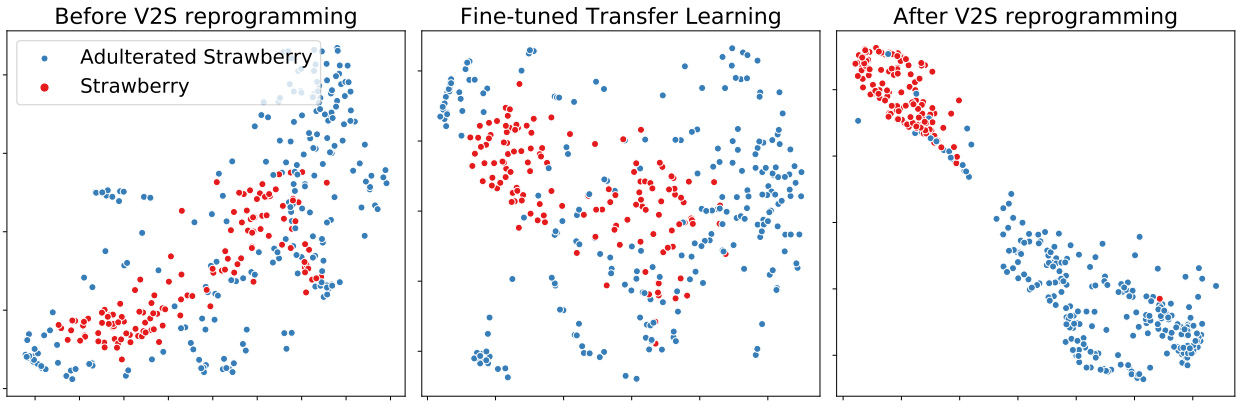


Figure 5. tSNE plots of the logit representations using the Strawberry training set (Holland et al., 1998) and V2S$_a$, for the cases of before and after V2S reprogramming, and fine-tuned transfer learning (TF$_a$).

tic neighbor embedding (tSNE) (Van der Maaten & Hinton, 2008) to visualize the logit representations of the Strawberry training set (Holland et al., 1998) for the cases of before and after reprogramming, and the transfer learning baseline (TF$_a$). As shown in Figure 5, after reprogramming tSNE results show a clear separation between the embeddings from different target classes, suggesting that V2S indeed learns meaningful and discriminative data representations to reprogram the pre-trained acoustic model for time series classification. On the other hand, the embedding visualization of transfer learning shows low class-wise separability.

5.6. Additional Discussion

In what follows, we provide additional discussion and insights for V2S reprogramming.

Many-to-one Label Mapping: The many-to-one mapping can be viewed as ensemble averaging of single-class outputs, a practical technique to improve classification for V2S reprogramming. We report ablation results (mean/medium accuracy (%)) using Table 2's setup controlled by q-to-1 V2S$_a$ mapping as follows: $\{q$=1: 89.21/94.17, q=2: 89.58/94.52, q=3: 89.86/94.99$\}$. These results show that many-to-one label mapping performs better results in our experiments.

Significance Testing: In Section 5, our results suggest that V2S as a *single* method can achieve or exceed SOTA on 20 out of 30 datasets obtained from *different* methods. We further run significant testing on two levels using their accuracies based on the 30 datasets from Table 2 — (i) V2S$_a$ v.s. SOTA numbers: p-value=0.0017; (ii) V2S v.s. FCN (Wang et al., 2017): p-value=0.0011, which indicate our results are significant.

Effect of Source Dataset: The effect of source dataset for V2S is captured by the source risk ϵ_S in Theorem 1, along with the representation alignment loss via SWD. We first evaluate different acoustic datasets to calculate their source test errors, where GSCv2 attains the smallest test error. The source test errors on $\{$GSCv2, TAU, AudioSet, ESC$\}$ are $\{0.1709, 0.1822, 0.1839, 0.1765\}$, and their SWD are reported in Appendix Table 5. Our theorem informs the performance of V2S on different source datasets. If other datasets can have smaller ϵ_S and SWD than GSCv2, we expect it to have better V2S performance.

Future Works: Our future works include a wider range of performance evaluations on different acoustic and speech models (e.g., those associated with lexical information) for V2S reprogramming, model reprogramming for low-resource speech processing, and extension to multivariate time series tasks. The proposed theory would also provide insights on analyzing the success of adversarial reprogramming in vision and language processing domains.

6. Conclusion

In this work, we proposed V2S, a novel approach to reprogram a pre-trained acoustic model for time series classification. We also developed a theoretical risk analysis to characterize the reprogramming performance. Experimental results on UCR benchmark showed superior performance of V2S, by achieving new (or equal) state-of-the-art accuracy on 20 out of 30 datasets. We also provided in-depth studies on the success of V2S through representation alignment, acoustic saliency map, and embedding visualization.

Acknowledgements

The authors would like to thank the comments and discussion from anonymous reviewers during the double-blind review process.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pp. 265–283, 2016.

Bagnall, A., Lines, J., Hills, J., and Bostrom, A. Time-series classification with cote: the collective of transformation-based ensembles. *IEEE Transactions on Knowledge and Data Engineering*, 27(9):2522–2535, 2015.

Cabello, N., Naghizade, E., Qi, J., and Kulik, L. Fast and accurate time series classification through supervised interval search. *ICDM 2020*, 2020.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *ACM Workshop on Artificial Intelligence and Security*, pp. 15–26, 2017.

Choi, K., Joo, D., and Kim, J. Kapre: On-gpu audio preprocessing layers for a quick implementation of deep neural network models with keras. In *Machine Learning for Music Discovery Workshop at 34th International Conference on Machine Learning*. ICML, 2017.

Cramer, J., Wu, H.-H., Salamon, J., and Bello, J. P. Look, listen, and learn more: Design choices for deep audio embeddings. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3852–3856. IEEE, 2019.

Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., and Keogh, E. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

Davis, L. M. *Predictive modelling of bone ageing*. PhD thesis, University of East Anglia, 2013.

de Andrade, D. C., Leo, S., Viana, M. L. D. S., and Bernkopf, C. A neural attention model for speech command recognition. *arXiv preprint arXiv:1808.08929*, 2018.

Elsayed, G. F., Goodfellow, I., and Sohl-Dickstein, J. Adversarial reprogramming of neural networks. In *International Conference on Learning Representations*, 2019.

Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. Transfer learning for time series classification. In *2018 IEEE international conference on big data (Big Data)*, pp. 1367–1376. IEEE, 2018.

Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4): 917–963, 2019.

Fritz, J. B., Elhilali, M., David, S. V., and Shamma, S. A. Auditory attention—focusing the searchlight on sound. *Current opinion in neurobiology*, 17(4):437–455, 2007.

Gao, B. and Pavel, L. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017.

Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 776–780. IEEE, 2017.

Geurts, P. Pattern extraction for time series classification. In *European conference on principles of data mining and knowledge discovery*, pp. 115–127. Springer, 2001.

Hambardzumyan, K., Khachatrian, H., and May, J. Warp: Word-level adversarial reprogramming. *arXiv preprint arXiv:2101.00121*, 2020.

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

Heittola, T., Mesaros, A., and Virtanen, T. Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions. *arXiv preprint arXiv:2005.14623*, 2020.

Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pp. 131–135. IEEE, 2017.

Hills, J., Lines, J., Baranauskas, E., Mapp, J., and Bagnall, A. Classification of time series by shapelet transformation. *Data mining and knowledge discovery*, 28(4):851–881, 2014.

Holland, J., Kemsley, E., and Wilson, R. Use of fourier transform infrared spectroscopy and partial least squares regression for the detection of adulteration of strawberry purees. *Journal of the Science of Food and Agriculture*, 76(2):263–269, 1998.

Hong, S., Xiao, C., Ma, T., Li, H., and Sun, J. Mina: Multilevel knowledge-guided attention for modeling electrocardiography signals. *arXiv preprint arXiv:1905.11333*, 2019.

Hu, H., Yang, C.-H. H., Xia, X., Bai, X., Tang, X., Wang, Y., Niu, S., Chai, L., Li, J., Zhu, H., et al. Device-robust acoustic scene classification based on two-stage categorization and data augmentation. *arXiv preprint arXiv:2007.08389*, 2020.

Hu, H., Yang, C.-H. H., Xia, X., Bai, X., Tang, X., Wang, Y., Niu, S., Chai, L., Li, J., Zhu, H., et al. A two-stage approach to device-robust acoustic scene classification. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 845–849. IEEE, 2021.

Kampouraki, A., Manis, G., and Nikou, C. Heartbeat time series classification with support vector machines. *IEEE Transactions on Information Technology in Biomedicine*, 13(4):512–518, 2008.

Kantorovich, L. and Rubinstein, G. On a space of completely additive functions. In *Vestnik Leningradskogo Universiteta*, volume 13 (7), pp. 52–59, 1958.

Kashiparekh, K., Narwariya, J., Malhotra, P., Vig, L., and Shroff, G. Convtimenet: A pre-trained deep convolutional neural network for time series classification. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2019.

Kaya, E. M. and Elhilali, M. Modelling auditory attention. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 372(1714):20160101, 2017.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

Kloberdanz, E. Reprogramming of neural networks: A new and improved machine learning technique. *ISU Master Thesis*, 2020.

Kolouri, S., Rohde, G. K., and Hoffmann, H. Sliced wasserstein distance for learning gaussian mixture models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3427–3436, 2018.

Langkvist, M., Karlsson, L., and Loutfi, A. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 42:11–24, 2014.

Lea, C., Vidal, R., Reiter, A., and Hager, G. D. Temporal convolutional networks: A unified approach to action segmentation. In *European Conference on Computer Vision*, pp. 47–54. Springer, 2016.

Lines, J., Davis, L. M., Hills, J., and Bagnall, A. A shapelet transform for time series classification. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 289–297, 2012.

Lines, J., Taylor, S., and Bagnall, A. Time series classification with hive-cote: The hierarchical vote collective of transformation-based ensembles. *ACM Transactions on Knowledge Discovery from Data*, 12(5), 2018.

Liu, S., Chen, P.-Y., Kailkhura, B., Zhang, G., Hero, A., and Varshney, P. K. A primer on zeroth-order optimization in signal processing and machine learning. *IEEE Signal Processing Magazine*, 2020.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

Neekhara, P., Hussain, S., Dubnov, S., and Koushanfar, F. Adversarial reprogramming of text classification neural networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5219–5228, 2019.

Peyré, G. and Cuturi, M. Computational optimal transport. arxiv e-prints. *arXiv preprint arXiv:1803.00567*, 2018.

Piczak, K. J. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pp. 1015–1018. ACM Press, 2025. ISBN 978-1-4503-3459-4. doi: 10.1145/2733373.2806390. URL http://dl.acm.org/citation.cfm?doid=2733373.2806390.

Ratanamahatana, C. A. and Keogh, E. Three myths about dynamic time warping data mining. In *Proceedings of the 2005 SIAM international conference on data mining*, pp. 506–510. SIAM, 2005.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Saon, G., Kurata, G., Sercu, T., Audhkhasi, K., Thomas, S., Dimitriadis, D., Cui, X., Ramabhadran, B., Picheny, M., Lim, L.-L., et al. English conversational telephone speech recognition by humans and machines. *Proc. Interspeech 2017*, pp. 132–136, 2017.

Schäfer, P. The boss is concerned with time series classification in the presence of noise. *Data Mining and Knowledge Discovery*, 29(6):1505–1530, 2015.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015.

Tsai, Y.-Y., Chen, P.-Y., and Ho, T.-Y. Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. In *International Conference on Machine Learning*, pp. 9614–9624, 2020.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Veale, R., Hafed, Z. M., and Yoshida, M. How is visual salience computed in the brain? insights from behaviour, neurobiology and modelling. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 372(1714): 20160113, 2017.

Vinod, R., Chen, P.-Y., and Das, P. Reprogramming language models for molecular representation learning. *arXiv preprint arXiv:2012.03460*, 2020.

Wang, Z., Yan, W., and Oates, T. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pp. 1578–1585. IEEE, 2017.

Warden, P. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

Wu, Y. and Lee, T. Enhancing sound texture in cnn-based acoustic scene classification. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 815–819. IEEE, 2019.

Xu, Y., Du, J., Dai, L.-R., and Lee, C.-H. A regression approach to speech enhancement based on deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(1):7–19, 2014.

Yang, C.-H., Qi, J., Chen, P.-Y., Ma, X., and Lee, C.-H. Characterizing speech adversarial examples using self-attention u-net enhancement. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3107–3111. IEEE, 2020.

Yang, C.-H. H., Qi, J., Chen, S. Y.-C., Chen, P.-Y., Siniscalchi, S. M., Ma, X., and Lee, C.-H. Decentralizing feature extraction with quantum convolutional neural network for automatic speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6523–6527. IEEE, 2021a.

Yang, C.-H. H., Siniscalchi, S. M., and Lee, C.-H. Pate-aae: Incorporating adversarial autoencoder into private aggregation of teacher ensembles for spoken command classification. *arXiv preprint arXiv:2104.01271*, 2021b.

Ye, R. and Dai, Q. A novel transfer learning framework for time series forecasting. *Knowledge-Based Systems*, 156: 74–99, 2018.

Zhang, D., Zuo, W., Zhang, D., and Zhang, H. Time series classification using support vector machine with gaussian elastic metric kernel. In *2010 20th International Conference on Pattern Recognition*, pp. 29–32. IEEE, 2010.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

Appendix

A. Proof of Lemma 1

For brevity we use $[K]$ to denote the integer set $\{1, 2, \ldots, K\}$. We have

$$\mathbb{E}_{x\sim\mathcal{D},\ x'\sim\mathcal{D}'}\|f(x)-f(x')\|_2 \overset{(a)}{=} \mathbb{E}_{x\sim\mathcal{D},\ x'\sim\mathcal{D}'}\|\eta(z(x))-\eta(z(x'))\|_2 \tag{5}$$

$$\overset{(b)}{=} \int_{x\sim\mathcal{D},\ x'\sim\mathcal{D}'} \|\eta(z(x))-\eta(z(x'))\|_2 \Phi_{\mathcal{D},\mathcal{D}'}(x,x')dxdx' \tag{6}$$

$$\overset{(c)}{=} \int_{x\sim\mathcal{D},\ x'\sim\mathcal{D}'} \|\eta(z(x))-\eta(z(x'))\|_2 \Phi_{\mathcal{D}}(x)\cdot\Phi_{\mathcal{D}'}(x')dxdx' \tag{7}$$

$$\overset{(d)}{\leq} \sqrt{K}\cdot\int_{x\sim\mathcal{D},\ x'\sim\mathcal{D}'} \max_{k\in[K]} |[\eta(z(x))]_k-[\eta(z(x'))]_k|\Phi_{\mathcal{D}}(x)\cdot\Phi_{\mathcal{D}'}(x')dxdx' \tag{8}$$

$$\overset{(e)}{\leq} 2\sqrt{K}\cdot \sup_{g:\mathbb{R}^K\mapsto\mathbb{R},\|g\|_{\text{Lip}}\leq 1} \mathbb{E}_{x\sim\mathcal{D}}[g(z(x))]-\mathbb{E}_{x'\sim\mathcal{D}'}[g(z(x'))] \tag{9}$$

$$\overset{(f)}{=} 2\sqrt{K}\cdot\mathcal{W}_1(\mu_z,\mu_z') \tag{10}$$

(a) follows the neural network model, (b) follows the definition of expectation, (c) follows the assumption of independent data drawing, and (d) follows that $\|x\|_2=\sqrt{\sum_i^d x_i^2}\leq\sqrt{d\cdot\max_i[x_i^2]}=\sqrt{d}\cdot\max_i|x_i|$, and thus $\|\eta-\eta'\|_2\leq\sqrt{K}\cdot\max_k|[\eta-\eta']_k|$. (e) holds by setting $k^+=\arg\max_{k\in[K]}[\eta(z(x))]_k-[\eta(z(x'))]_k$ and $k^-=\arg\max_{k\in[K]}[\eta(z(x'))]_k-[\eta(z(x))]_k$. Then by definition $\max_{k\in[K]}|[\eta(z(x))]_k-[\eta(z(x'))]_k|\leq[\eta(z(x))]_{k^+}-[\eta(z(x'))]_{k^+}+[\eta(z(x'))]_{k^-}-[\eta(z(x))]_{k^-}$. We further make the following three notes: (i) $[\eta(z(x))]_{k^+}-[\eta(z(x'))]_{k^+}\geq 0$ and $[\eta(z(x))]_{k^-}-[\eta(z(x'))]_{k^-}\geq 0$; (ii) $|a|=\max\{a,-a\}$, and if $a,b\geq 0$, $\max\{a,b\}\leq a+b$; (iii) There exist at least one k such that $[\eta(x)]_k-[\eta(x)]_k\geq 0$. One can use proof by contradiction to show (iii) is true. If $[\eta(x)]_k-[\eta(x)]_k<0$ for every k, then summing over k we get a contradiction that $1<1$. Therefore,

$$\int_{x\sim\mathcal{D},\ x'\sim\mathcal{D}'} \max_{k\in[K]} |[\eta(z(x))]_k-[\eta(z(x'))]_k|\Phi_{\mathcal{D}}(x)\cdot\Phi_{\mathcal{D}'}(x')dxdx' \tag{11}$$

$$\leq \int_{x\sim\mathcal{D},\ x'\sim\mathcal{D}'} \left([\eta(z(x))]_{k^+}-[\eta(z(x))]_{k^-}+[\eta(z(x'))]_{k^-}-[\eta(z(x'))]_{k^+}\right)\Phi_{\mathcal{D}}(x)\cdot\Phi_{\mathcal{D}'}(x')dxdx' \tag{12}$$

$$=\mathbb{E}_{x\sim\mathcal{D}}[[[\eta(z(x))]_{k^+}-[\eta(z(x))]_{k^-}]-\mathbb{E}_{x'\sim\mathcal{D}'}[[\eta(z(x'))]_{k^+}-[\eta(z(x'))]_{k^-}] \tag{13}$$

$$\leq 2\cdot \sup_{g:\mathbb{R}^K\mapsto\mathbb{R},\|g\|_{\text{Lip}}\leq 1} \mathbb{E}_{x\sim\mathcal{D}}[g(z(x))]-\mathbb{E}_{x'\sim\mathcal{D}'}[g(z(x'))] \tag{14}$$

where $\|g\|_{\text{Lip}}$ is defined as $\sup_{x,x'}|g(x)-g(x')|/\|x-x'\|_2$, and we use the fact that $[\eta(z)]_k$ is 1-Lipschitz for any $k\in[K]$ (Gao & Pavel, 2017) (so $[\eta]_{k^+}-[\eta]_{k^-}$ is 2-Lipschitz). Finally, (f) follows the Kantorovich-Rubinstein theorem (Kantorovich & Rubinstein, 1958) of the dual representation of the Wasserstein-1 distance.

B. Proof of Theorem 1

First, we decompose the target risk function as

$$\ell_{\mathcal{T}}(x_t+\delta^*,y_t) \overset{(a)}{=} \ell_{\mathcal{S}}(x_t+\delta^*,y_s) \tag{15}$$

$$\overset{(b)}{=} \|f_{\mathcal{S}}(x_t+\delta^*)-y_s\|_2 \tag{16}$$

$$\overset{(c)}{=} \|f_{\mathcal{S}}(x_t+\delta^*)-f_{\mathcal{S}}(x_s)+f_{\mathcal{S}}(x_s)-y_s\|_2 \tag{17}$$

$$\overset{(d)}{\leq} \underbrace{\|f_{\mathcal{S}}(x_t+\delta^*)-f_{\mathcal{S}}(x_s)\|_2}_{A}+\underbrace{\|f_{\mathcal{S}}(x_s)-y_s\|_2}_{B} \tag{18}$$

(a) is based on Assumption 3, (b) is based on the definition of risk function, (c) is by subtracting and adding the same term $f_{\mathcal{S}}(x_s)$, and (d) is based on the triangle inequality.

Note that by Assumption 1, $\mathbb{E}_{\mathcal{D}_\mathcal{S}} B = \mathbb{E}_{\mathcal{D}_\mathcal{S}}[\ell(x_s, y_s)] = \epsilon_\mathcal{S}$. Next, we proceed to bound $\mathbb{E}_{\mathcal{D}_\mathcal{S},\mathcal{D}_\mathcal{T}} A \triangleq \mathbb{E}_{x_s\sim\mathcal{D}_\mathcal{S},x_t\sim\mathcal{D}_\mathcal{T}} A$. Using Lemma 1, we have

$$\mathbb{E}_{\mathcal{D}_\mathcal{S},\mathcal{D}_\mathcal{T}} A \leq 2\sqrt{K} \cdot \mathcal{W}_1(\mu(z_\mathcal{S}(x_t + \delta^*)), \mu(z_\mathcal{S}(x_s)))_{x_t\sim\mathcal{D}_\mathcal{T},\, x_s\sim\mathcal{D}_\mathcal{S}} \tag{19}$$

Finally, take $\mathbb{E}_{\mathcal{D}_\mathcal{S},\mathcal{D}_\mathcal{T}}$ on both sides of equation (18) completes the proof.

C. Pre-Trained Model Studies

We provide advanced studies over different pre-trained acoustic architectures and the associated time series classification performance. In particular, we select the models below, which have attained competitive performance tested on Google Speech Commands version 2 dataset (Warden, 2018) or shown cutting-edge performance (ResNet (He et al., 2016)) in the acoustic scene (VGGish (Hershey et al., 2017)) and time series classification (TCN (Lea et al., 2016)). These models will be compared with V2S$_a$ (recurrent Attention (de Andrade et al., 2018)) and V2S$_u$ (V2S$_a$ enhanced by U-Net (Yang et al., 2021a)) used in the main paper.

ResNet: Deep residual network (He et al., 2016) (ResNet) is a popular deep architecture to resolve the gradient vanish issues by passing latent features with a residual connection, and it has been widely used in acoustic modeling tasks. We select a 34-layer ResNet model training from the scratch for V2S (denoted as V2S$_r$), which follows the identical parameter settings in (Hu et al., 2020; 2021) for reproducible studies.

VGGish: VGGish (Hershey et al., 2017) is a deep and wide neural network architecture with multi-channel convolution layers, which has been proposed for speech and acoustic modeling. VGGish is also well-known for the large-scale acoustic embedding studies with Audio-Set (Gemmeke et al., 2017) from 2 million Youtube audios. We use the same architecture and train two models: (i) training from scratch (denoted as V2S$_v$) and (ii) selecting an Audio-Set pretrained VGGish and then fine-tuning (denoted as V2S$_p$) on the Google Speech commands dataset for V2S.

Temporal Convolution Network (TCN): TCN (Lea et al., 2016) is an efficient architecture using temporal convolution with causal kernel for sequence classification tasks. We select the TCN architecture and train it from scratch as a baseline for V2S (denoted as V2S$_t$).

OpenL3: OpenL3 (Cramer et al., 2019) is a much recent embedding method with a deep fusion layer for acoustic modeling. We use pretrained OpenL3 embeddings and a dense layer for classification as another baseline for V2S (denoted as V2S$_o$).

C.1. V2S Performance and Sliced Wasserstein Distance

Table 4 shows different neural architectures for acoustic modeling to be used with the proposed V2S method, where acoustic models are pre-trained with the Google Speech Commands dataset (Warden, 2018) version two with 32 commands (denoted as GSCv2.) From the first three rows of Table 4, we observe the recurrent attention models and TCN perform better in mean prediction accuracy, validation loss of the source task. For the target task (same as Table 2), recurrent attention models (V2S$_a$ and V2S$_u$) attain the best performance.

Table 4. V2S ablation studies with different pre-trained acoustic models.

Model	V2S$_a$	V2S$_u$	V2S$_r$	V2S$_v$	V2S$_p$	V2S$_t$	V2S$_o$
Parameters (↓)	**0.2M**	0.3M	1M	62M	62M	1M	4.7M
Source Acc. (↑)	96.90	96.92	96.40	95.40	95.19	**96.93**	92.34
Source Loss (↓)	**0.1709**	0.1734	0.1786	0.1947	0.1983	0.1756	0.2145
Mean SWD (↓)	**1.829**	1.873	1.892	4.713	4.956	1.901	5.305
Mean Target Acc. (↑)	**89.91**	87.92	87.22	67.12	63.23	86.45	60.34
Target MPCE (↓)	**2.03**	2.10	2.23	33.4	38.3	2.34	41.34

Both VGGish based architectures (V2S$_v$ and V2S$_p$) show degraded performance on the target tasks prediction, which can be explained by the recent findings (Kloberdanz, 2020) on the degraded performance of using VGG (Simonyan & Zisserman, 2015) and MobileNet-V2 (Sandler et al., 2018) based "wide" and deep convolutional neural architectures for reprogramming visual models. These findings could be also explained in the sense that the wide neural architectures fail to adapt the source

domain distribution according to the sliced Wasserstein distance (SWD) results (fourth row in Table 4), as indicated by Theorem 1. We observe that using the pretrained models associated with higher source accuracy does not always guarantee higher target accuracy (e.g, V2S$_t$ and V2S$_u$)

D. Additional Ablation Studies

Based on the discussion in Section C, we further select three efficient V2S models with different model capacity (0.2M/1M/4.7M), including V2S$_a$, V2S$_r$, and V2S$_t$, to study the mean target accuracy in different training settings and to provide some insights into effective design of V2S models.

D.1. Pretrained Models from Different Dataset

In the previous model reprogramming studies (Tsai et al., 2020; Elsayed et al., 2019), little has been discussed regarding the effectiveness of using different datasets to train the pretrained models. We study three other public acoustic classification benchmark datasets (source tasks), (1) TAU Urban Acoustic Scenes 2020 Mobile (Heittola et al., 2020) (denoted as TAU-UAC), from the annual IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE), (2) AudioSet using in (Hershey et al., 2017), and (3) ESC-50 (Piczak, 2025), a dataset for environmental sound classification, for providing a preliminary V2S study. We extract acoustic features by Mel-spectrogram using Kapre audio layer following the same resolution and frequency setup in (Yang et al., 2020; 2021b) for a fair and reproducible comparison.

As shown in Table 5, the V2S models pretrained from GSCv2 show a higher mean prediction accuracy and lower SWD than the other models, which could be due to the shorter sample length (less than one second) of its source acoustic inputs.

Table 5. V2S performance (mean prediction accuracy) on time series classification (same as Table 2) with different pretrained neural acoustic models and the mean sliced Wasserstein distance for each dataset.

Pretrained Dataset	GSCv2	TAU	AudioSet	ESC
# Training Samples	105k	13.9k	2.08M	2k
# Output Classes	35	10	527	50
Audio length per sample clips	1 sec.	10 sec.	10 sec.	5 sec.
Mean Target Acc. w/ V2S$_a$ (↑)	**89.91**	82.61	80.68	84.48
Mean Target Acc. w/ V2S$_r$ (↑)	87.22	83.57	79.96	83.05
Mean Target Acc. w/ V2S$_t$ (↑)	86.45	80.1	81.81	84.58
Mean SWD per dataset (↓)	**1.874**	2.267	2.481	2.162

D.2. Different V2S Mapping Settings

In (Tsai et al., 2020), frequency mapping techniques show improved performance for black-box (e.g., zeroth order gradient estimation (Chen et al., 2017; Liu et al., 2020) based) adversarial reprogramming models for image classification. We also follow the setup in (Tsai et al., 2020) to compare the many-to-one frequency mapping and the many-to-one random mapping for time series classification. However, the frequency mapping based V2S results show equal or slightly worse (-0.013%) mean prediction accuracy and WSD (+0.0028) performance with 100 runs, which may be owing to the differences of the dimensions and scales between the tasks of image and time series reprogramming.

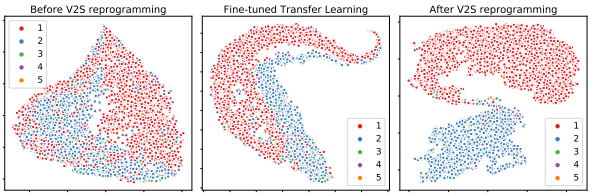
D.3. More tSNE Visualization

We provide more tSNE visualization over different test datasets to better understand the embedding results of V2S models discussed in Section 5.5. In Figure 6 (a) to (e), the reprogrammed representations (rightmost side) show better disentangled results in both 2D and 3D tSNE plots.

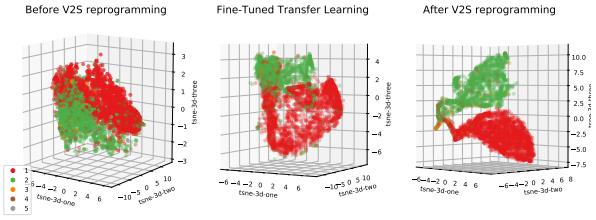
D.4. Hardware Setup and Energy Cost Discussion

We use Nvidia GPUs (2080-Ti and V100) for our experiments with Compute Unified Device Architecture (CUDA) version 10.1. To conduct the results shown in Table 2, it takes around 40 min to run 100 epochs (maximum) with a batch size 32 for each time series prediction dataset considering the hyper-parameters tuning (e.g., dropout rate) described in Section 5.2

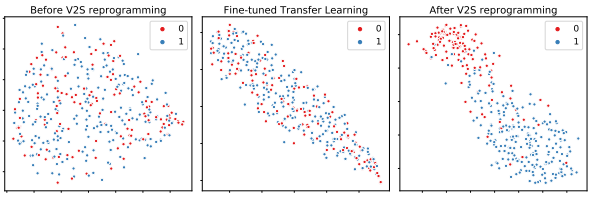
of the main paper. In total, the experiments presented (30 datasets and its ablation studies) in this paper took around 120 computing hours with a 300W power supplier. As another advantage, the V2S reprogramming techniques freeze pretrained neural models and only used a reprogramming layer for training new tasks. The proposed method could potentially recycle well-trained models for an additional task to alleviate extra energy costs toward deploying responsible ML systems.



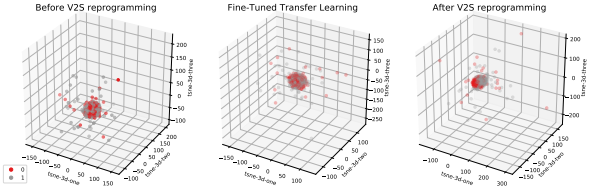
(a) Task: ECG 5000 with 2D tSNE



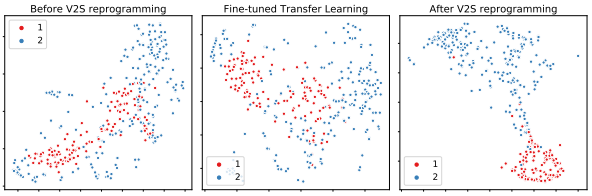
(b) Task: ECG 5000 with 3D tSNE



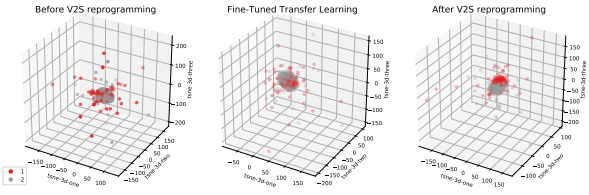
(c) Task: HandOutlines with 2D tSNE.



(d) Task: HandOutlines with 3D tSNE.



(e) Task: Strawberry 2D tSNE.



(f) Task: Strawberry 3D tSNE

Figure 6. More tSNE visualization. Numbers in the legend are class label indices.