Compositionality in (Clinical) Time Series

Anonymous authors
Paper under double-blind review

Abstract

This work investigates whether time series of clinical measurements can be understood as being generated by meaningful physiological states whose succession follows compositional principles. Since there is no obvious definition of elementary components and composition rules in time series, we approach this task by first conceptualizing compositionality in time series data as a property of the data generation process, and then study data-driven learning procedures that can revert this process by deconstructing times series into elementary states and composition rules. Our empirical pipeline involves a symbolization of time series and a data augmentation procedure to synthesize full time series in a compositional manner. We propose two empirically testable conditions for compositionality that are motivated from a domain adaptation perspective. Both tests infer the similarity of the distributions of clinical time series and of compositionally synthesized data from the expected risk of time series forecasting models trained and tested on original and synthesized data. Our experimental results show that the test set performance achieved by training on compositionally synthesized data is comparable to training on original clinical time series data, and that evaluation of models on compositionally synthesized test data shows similar results to evaluating on original test data. In both experiments, performance based on compositionally synthesized data by far surpasses that based on synthetic data that were created by randomization-based data augmentation. This work sheds some light on the compositional nature of clinical time series and introduces a general theoretically motivated framework work to empirically assess the compositionality of an unspecified data-generating process.

1 Introduction

Compositionality describes the systematic capacity of a system to generate an unbounded number of valid outputs based on novel combinations from a finite set of elementary components. This concept arose independently in logic, natural language processing, and computer science, beginning in the 19th century (see Partee (1984), Janssen (2012), or Szabó (2020) for historical overviews). Compositionality is considered one of the main pillars of the human ability to generalize to new tasks and situations (Lake et al., 2017), and it currently experiences a renaissance as compositional generalization — the ability to systematically generalize to test data that are composed from known components seen in novel combinations — in various machine learning tasks (see Lin et al. (2023) for an overview). The standard examples for compositionality in human and machine intelligence are natural language and vision, for example, the capacity to build an infinite number of sentences from a finite vocabulary, or the composition of images from elementary concepts of color, position, and shape. Our goal is to investigate whether the concept of compositionality can be applied to the less obvious domain of time series, exemplified by the task of clinical time series forecasting.

Our notion of compositionality in time series data takes inspiration from the formalization of a general compositional data generation process given by Wiedemer et al. (2023). We use this concept to motivate an understanding of clinical time series as being generated by meaningful physiological states whose succession follows compositional principles, and to motivate a data-driven pipeline that reverts this process to empirically detect elementary states and composition rules. Our empirical pipeline uses representation learning, in particular clustering of learned representations of time series subsequences (Ghaderi et al., 2023; Ma et al., 2019), to induce latent classes representing meaningful physiological states, for example, healthy or unhealthy states of certain organ systems. Mapping time series to symbolic representations makes them amenable to compositional methods developed for natural language processing (NLP). Here we apply a simple but entirely data-driven approach to inducing compositional structure from symbolic representations of time series (Andreas, 2020). This algorithm implements the distributional principle (Firth, 1957) to exchange subsequences that occur in the same context to yield other valid sequences.

In contrast to disentangled representation learning in vision, the latent factors are unknown for time series, thus we cannot formalize a test for compositional generalization as a direct reconstruction of the data generation process. Instead, we generate synthetic data in a compositional manner, and use the similarity of original time series to synthetic data to infer the compositionality of the data generation process of original time series. Our test criteria are motivated in domain adaptation theory (Ben-David et al., 2006; 2010b;a; Ben-David & Urner, 2014). We exploit the assumption that successful domain adaptation requires a small distance between source distribution (in our case, the distribution underlying compositionally synthesized data) and target distribution (in our case, original time series data), to infer a small distance between the underlying distributions of synthesized and original data from empirically testing the success of domain adaptation from synthetic to original data.

Our first test evaluates compositionally synthesized time series by analyzing their utility for the training of a time series forecasting (TSF) model. Our experiments demonstrate comparable test set performance of models trained on compositionally synthetic data to models trained on the original data for MIMIC-III (Johnson et al., 2016) and eICU (Pollard et al., 2018)) data sets. Our second test compares the use of compositionally synthesized data against original time series data as test data in TSF tasks. Again, both test sets yield a similar test set performance for a TSF model trained on original time series data. Furthermore, we find that compositionally synthesized data is much closer to the original data than synthetic data created by a non-compositional data augmentation algorithm (Yun et al., 2019).

In sum, the contributions[1] of our work are as follows: We present an analysis based on symbolic representations and compositional structure that allows an understanding of clinical time series as symbols that are emitted by an ordered sequence of physiological states. The underlying algorithm allows us to create synthetic training and test data that are on par with the original data in a real-world clinical time series forecasting task. This circumvents the notorious problem of sparse and low-resource data settings in clinical TSF. Furthermore, we present empirically testable criteria for compositionality rooted in domain adaptation theory, allowing broader claims on the compositionality of the data generation process underlying general time series data.

2 Related Work

Symbolic Compositional Structure in NLP. A connection of our work to NLP can be drawn by viewing natural language sentences as discrete time series of symbols, and by considering the task of next word prediction in language modeling as a TSF task. The central models developed in symbolic NLP — the dominant paradigm in 20th century NLP — were all explicit symbolic generative processes allowing to construct an infinite number of sentences from a finite alphabet (vocabulary) and a finite recursive device (grammar)[2]. Since NLP research has long departed from the symbolic paradigm, the central question in current research on compositional generalization in NLP is consequently an investigation of the compositional skills of non-symbolic neural network architectures. One research strand on compositionality in non-symbolic NLP focuses on the creation of benchmark datasets to evaluate the compositional generalization abilities of neural networks (Lake & Baroni (2018); Keysers et al. (2020); Kim & Linzen (2020), inter alia). Most such datasets are based on the NLP task of semantic parsing, where the components and composition rules are obvious and can be clearly defined. Compositionality is then quantified by measuring accuracy on test sets with a similar component distribution, but different compound distribution. Another research strand focuses on directly injecting a compositional inductive bias into neural sequence models (Russin et al. (2020); Huang et al. (2024); Sartran et al. (2022), inter alia). Our approach directly builds on the algorithm of Andreas

---

[1] Upon acceptance of the paper, code to reproduce the experiments will be made publicly available at https://anon-url.com

[2] Symbolic approaches dominated the NLP fields of syntax and semantics, starting with Chomsky (1957) and Montague (1970), respectively, and are still relevant in formal language theory in computer science (starting with Chomsky (1959)).

(2020) that aims to provide a compositional inductive bias to state-of-the-art sequence learning models by adding compositionally generated data to their training sets.

Disentangled Representation Learning in Vision. In disentangled representation learning, real-world observations x (for example, images) are considered to be generated in a two-step process: First, a multivariate latent random variable z (consisting of semantically meaningful factors describing variation between observations, for example, position, color and shape of objects in an image) is sampled from a distribution $P(z)$, then an observation point x (for example, an image composed of several objects) is sampled from $P(x|z)$. The goal of this framework is to learn disentangled representations $r(x)$ that separate informative factors such that a change in a single latent factor z_i leads to a change in a single factor in the learned representation $r(x)$ (Bengio et al., 2013). The state-of-the-art models in disentangled representation learning in vision are auto-encoders that directly aim to reconstruct the generative factors of variation (Higgins et al. (2017); Montero et al. (2021); Xu et al. (2022), inter alia). The generalization abilities of these models are usually tested by the task of reconstructing compositionally generated test data that include combinations of generative factors that were not seen during training. However, it has been shown that unsupervised disentangled representation learning is impossible without inductive biases on both models and data (Locatello et al., 2019), and that increased disentanglement does not increase generalization capabilities, especially if one moves away from a simple artificial data set to real-world data (Schott et al., 2022). Furthermore, in addition to learning the factors of variation, it is also necessary to understand the compositional mechanisms that combine objects in new ways (Montero et al., 2022). Wiedemer et al. (2023) formalize compositionality as a property of the data generation process, assuming the composition function, as well as the latent description of the observations to be known, effectively reducing the learning task to a reconstruction of the component functions. In contrast to this work, we cannot formalize compositional generalization as a direct reconstruction problem since neither latent factors nor the composition function are known for time series. Instead, we indirectly test compositional generalization by evaluating the utility of compositionally synthesized data as training and test data in real-world TSF tasks.

Domain Adaptation. Our work takes crucial inspiration from the work of Ben-David et al. (2006; 2010b;a); Ben-David & Urner (2014) to motivate empirically testable criteria for compositionality in domain adaptation theory. Starting from theoretical results that identify necessary and sufficent conditions for a successful DA, we created two experimental setups that allow us to infer a small distance between synthetic and original data based on the distance between expected risks estimates. Furthermore, our work shows how to generate a synthetic data distribution from an original sample with the properties of matching the original distribution on the level of elementary components, but differing from it at the level of compounds. Our results can be seen as strong hints at the potential of a formal study of compositional data synthesization, with the goal of a better understanding of its generalization properties.

Data Augmentation. Compositional data augmentation is a recent research area in NLP that aims to supply a compositional bias to state-of-the-art neural networks by using a compositional data generation process to augment neural network training data (Andreas (2020); Akyürek et al. (2021); Qiu et al. (2022), inter alia). While data augmentation in NLP can build on clearly defined elementary components for which composition rules need to be learned, data augmentation in vision is mostly based on non-compositional randomization processes where multiple images are mixed to create new examples (see Cao et al. (2024) for an overview). Such mixed-based data augmentation approaches have been successfully applied to the area of (physiological) time series data (Guo et al., 2023; Yang & Desell, 2022), and have been interpreted as implicit regularization techniques (Yun et al., 2019; Zhang et al., 2018). Our work provides empirical evidence for an advantage of augmentation techniques that mimic a compositional data generation processes over randomization-based data augmentation.

Symbolic Representation Learning. Research on transforming real-valued time series into symbolic representations has a long history (see, for example, Williams (2004) for an overview), where the central application is the analysis of dynamical systems (Lind & Marcus, 1995). Since the number of possible symbol assignments grows exponentially with the dimension of the time series, the traditional approach of partitioning the multidimensional phase space spanned by the input variables of a time series into finitely many pieces and then labeling each partition by a specific symbol is only feasible for very low dimensional time series. This problem is overcome by representation learning approaches that first map time series into an embedding space (whose dimensionality can be controlled), where clustering methods are then applied to partition the space (Ma et al., 2019; Ghaderi et al., 2023). We employ the latter techniques to learn a symbolic vocabulary that contains the elementary components of our compositional data synthesization process, and compare it to randomized version of traditional symbolic dynamics.

3 Compositionality in Time Series

3.1 Compositional Data Generation

We conceptualize compositionality as a property of the data generating process, following an algebraic formalization of compositionality (Montague, 1970; Partee, 1984; Szabó, 2020).

Definition 1. (Compositional data generation.) Let $(\mathcal{Z}, C_{\mathcal{Z}})$ be an algebraic structure of latent states where $C_{\mathcal{Z}}$ is called the latent state composition function, and let$(\mathcal{X}, C_{\mathcal{X}})$ be an algebraic structure (of the same type) of observations where $C_{\mathcal{X}}$ is called the observation composition function. Furthermore, let $\varphi\colon \mathcal{Z} \to \mathcal{X}$ be a homomorphism that maps latent states to observations. We call a data generating process $f : \mathcal{Z} \to \mathcal{X}$ that satisfies

$$f(z_1, \dots, z_K) = \varphi(C_{\mathcal{Z}}(z_1, \dots, z_K)) = C_{\mathcal{X}}(\varphi(z_1), \dots, \varphi(z_k)) \tag{1}$$

a *compositional data generation process.*

The data generating process of clinical time series can be understood as compositional by the following definition. Let latent states be physiological states of patients, $C_{\mathcal{Z}}$ be the progression of a patient through these states, observations be clinical measurements, and $C_{\mathcal{X}}$ be the temporal regularities within time series. Then Equation 1 can be interpreted as follows:

> The clinical assessment of a sequence of physiological states is a function of the clinical assessments of its constituents and the way they are sequentially ordered.

The right-hand side form of Equation 1 can be illustrated by the generative process shown in Figure 1. This process starts from physiological states of patients, which emit observable components from the space of clinical measurements, which are ordered over time into a full sequence of clinical measurements. The left-hand side of Equation 1 motivates a deconstruction of this process into elementary representations of latent states, which are composed to a sequence of latent states that is finally mapped to a sequence of observations.

3.2 Empirical Deconstruction and Compositional Data Synthesization

The conceptualization of compositionality as given in Definition 1 is crucially dependent on the homomorphism φ that preserves compositionality. While the right-hand side of Equation 1 characterizes a compositional data generation process, the left-hand side motivates a deconstruction of this process as a composition of latent states. This deconstruction process builds the basis of the experimental work presented in this paper and is illustrated in Figure 2: Starting from input data of multivariate clinical time series, we extract a real-valued vector representation of subsequences from the hidden states of a TSF model, which is then fed into a k-means clustering algorithm (Lloyd, 1982) that allows us to assign a symbolic representation to time series based on the cluster membership of its subsequences. These symbolic representations are the input for an entirely data-driven compositional data augmentation algorithm (Andreas, 2020) that results in an implicit set of rules allowing us to synthesize novel time series in a compositional manner. Each component of this pipeline will be described in more detail below.

3.3 A Domain Adaptation Perspective on Compositionality

Viewing the above sketched compositional data augmentation process from a domain adaptation (DA) perspective allows us to arrive at two empirically testable conditions for compositionality. These tests exploit

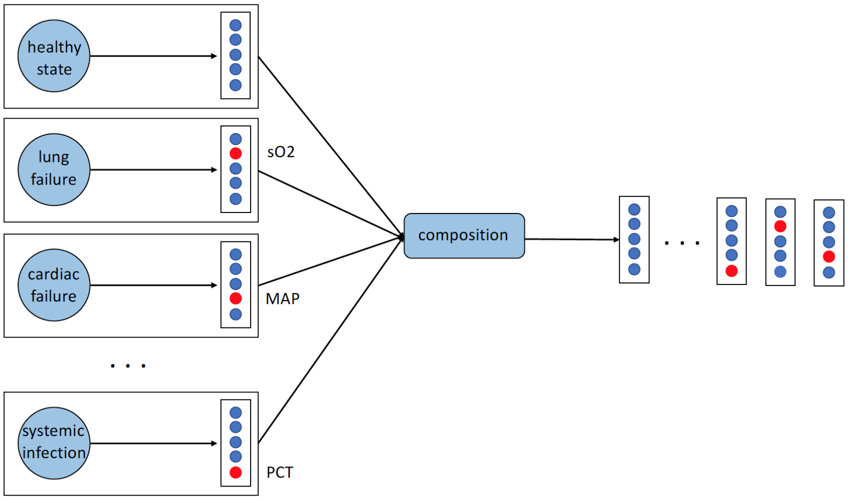


Figure 1: Conceptualization of compositional data generation process for time series of clinical variables: Physiological states are mapped to observations of clinical measurements which are composed into a full time series. Healthy physiological states are realized by observations consisting of default measurements of vital signals or lab measurements. Physiological states representing a failure of organ systems like hearth or lung are realized by observation vectors where the measurements of mean arterial pressure (MAP) or oxygen saturation (sO2) are out of a range defined as healthy, or a systemic infection causing an elevated procalcitonin (PCT) measurement. The clinical assessment of this example can be interpreted as a clinical time series of a septic patient with multiple organ system failure, starting with healthy measurements, which are followed by indicators of lung failure and cardiac failure, consequent to a indicators of a systemic infection.

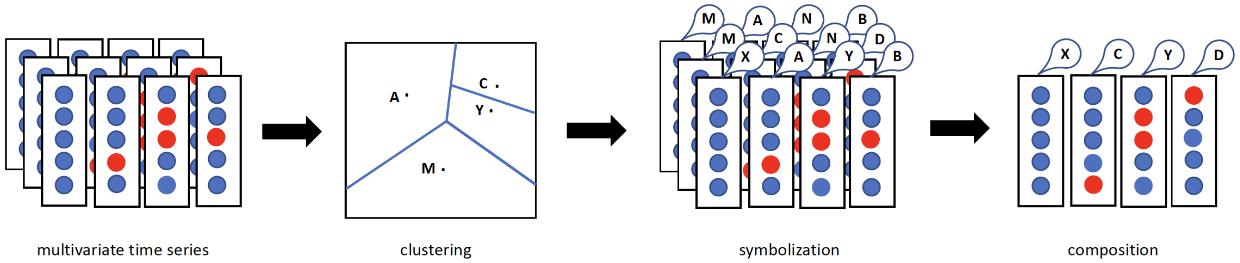


Figure 2: Empirical deconstruction of multivariate time series for compositional data synthesization: First, subsequences of time series are clustered and each cluster is assigned a symbolic label. Based on symbolic representations of time series sub-sequences, compositional data augmentation algorithms can be applied to synthesize full time series in a compositional manner.

the fact that the training and test distribution of a model are not identical in DA scenarios. For both of our empirical tests, one of these distributions will be synthetically created by a compositional algorithm, and the other distribution will be the original distribution whose compositional status is unknown. The compositionality of the data generation process underlying the original time series data can then be inferred from successful DA from compositionally synthesized to original data.

Our first test is based on theoretical results established by Ben-David et al. (2006; 2010b;a), which provide two jointly necessary and sufficient conditions for successful domain adaptation, one of which is a large enough similarity between training and target distributions. We leverage these results to motivate an empirically testable criterion based on the following rationale. We apply a learning algorithm to both the original training data and compositionally synthesized training data and evaluate the trained models on the original test data. If the estimated expected risks are identical, DA is considered successful and according to the results established by Ben-David et al. (2006; 2010b;a), we can conclude that both distributions must be identical. Thus, we can conclude that the original data generating process must be compositional.

Our second test is also grounded in the theory of DA and leverages a new result that extends an observation by Ben-David & Urner (2014). Our Theorem 1 relates the proximity of the expected risks of a model on two data distributions to the proximity of these distributions. We leverage this theorem to motivate a second empirically testable criterion based on the following rationale. We evaluate a trained learner on both the original test set and a compositionally synthesized test set, and estimate the ratio of the corresponding expected risks. As established by Theorem 1, the magnitude of this ratio allows drawing a conclusion about the proximity between the synthetic and original distributions. If this ratio is one, both distributions are identical, and hence the original data generating process must be compositional.

In the following, we briefly summarize the theoretical concepts of DA theory, especially the necessary and sufficient conditions for successful DA.

Let $\mathcal{X}$ be a domain, $\mathcal{Y}$ be a co-domain, and $\mathcal{P}$ and $\mathcal{Q}$ be distributions over $\mathcal{X} \times \mathcal{Y}$. Furthermore, let $\mathcal{P}_{\mathcal{X}}$ and $\mathcal{Q}_{\mathcal{X}}$ denote the respective marginal distributions on $\mathcal{X}$. Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a hypothesis class of functions h, and let $\ell\left(y, \hat{y}\right)$ be a loss function on the target labels y and the predicted labels $\hat{y} = h(x)$. We can now define the concepts of a domain adaptation learner (Definition 2), and a concept quantifying domain adaptation learnability (Definition 3).

Definition 2. (Domain adaptation learner.) We call a learning algorithm A that receives training samples from $\mathcal{Q}$ but whose expected risk is evaluated on $\mathcal{P}$ a *(conservative) domain adaptation learner*.

Definition 3. ((ϵ, δ)-learnability.) Let $\mathcal{P}$ and $\mathcal{Q}$ be distributions on $\mathcal{X} \times \mathcal{Y}$ with common support, $\mathcal{H}$ a hypothesis class, and A be a domain adaptation learner. We say that A *(ϵ, δ)-learns $\mathcal{P}$ from $\mathcal{Q}$ relative to $\mathcal{H}$*, if for chosen $\epsilon, \delta > 0$ there exists an $n \in \mathbb{N}$ such that when provided a training sample T of size n obtained from $\mathcal{Q}$, with probability of at least $1 - \delta$ (over the sample space), the expected risk with respect to $\mathcal{P}$ of the obtained classifier $h_T \coloneqq A\left(T\right)$ does not exceed the expected risk $\mathbb{E}_{\mathcal{P}}\left[\mathcal{H}\right] \coloneqq \inf_{h \in \mathcal{H}} \mathbb{E}_{\mathcal{P}}\left[\ell(y, h(x))\right]$ of the best classifier on $\mathcal{P}$ by more than ϵ:

$$\Pr_{T \sim \mathcal{Q}^n}\left(\mathbb{E}_{\mathcal{P}}\left[\ell(y, h_T(x))\right] - \mathbb{E}_{\mathcal{P}}\left[\mathcal{H}\right] \leq \epsilon\right) \geq 1 - \delta.$$

Obviously DA works well if $\mathbb{E}_{\mathcal{P}}\left[\ell(y, h_T(x))\right] - \mathbb{E}_{\mathcal{P}}\left[\mathcal{H}\right]$ can be bounded by a small ϵ, with high probability (small δ) for feasible sample sizes n.

A further popular assumption in the context of DA is the covariate shift assumption:

Definition 4. (Covariate shift assumption.) Let $\mathcal{P}$ and $\mathcal{Q}$ be distributions on $\mathcal{X} \times \mathcal{Y}$ with common support. Then $\mathcal{P}$ and $\mathcal{Q}$ satisfy the covariate shift assumption if the conditional distributions $\mathcal{P}_{\mathcal{Y}|\mathcal{X}}$ and $\mathcal{Q}_{\mathcal{Y}|\mathcal{X}}$ are identical.

The covariate shift assumption is a rather weak requirement that only demands that the stochastic relation between x and y be identical for $\mathcal{Q}$ and $\mathcal{P}$. This assumption is only a necessary, but not a sufficient, condition for a successful DA. This fact is exemplified by the following thought experiment. Let us assume that we trained a model on $\mathcal{Q}$, and that the difference between $\mathcal{Q}$ and $\mathcal{P}$ is such that $\mathcal{Q}_{\mathcal{X}}$ places a lot of mass on inputs that the learner predicts well, and less on inputs where the learner performs poorly. In addition, assume that the situation is exactly the opposite for $\mathcal{P}_{\mathcal{X}}$. Then the expected risk with respect to $\mathcal{Q}$ will be small, but the expected risk with respect to $\mathcal{P}$ will be large. This thought experiment illustrates the need to put a constraint on the difference between $\mathcal{P}_{\mathcal{X}}$ and $\mathcal{Q}_{\mathcal{X}}$. Since the seminal work of Ben-David et al. (2006), it is common to express this difference in terms of the so-called $\mathcal{A}$-distance:

Definition 5. ($\mathcal{A}$-distance.) Let $\mathcal{P}_{\mathcal{X}}$ and $\mathcal{Q}_{\mathcal{X}}$ be two distributions on $\mathcal{X}$ and $\mathcal{A} \subseteq 2^{\mathcal{X}}$ such that each set in $\mathcal{A}$ is measurable with respect to both distributions. Then the $\mathcal{A}$-distance between $\mathcal{P}_{\mathcal{X}}$ and $\mathcal{Q}_{\mathcal{X}}$ is

$$\mathrm{d}_{\mathcal{A}}\left(\mathcal{P}_{\mathcal{X}}, \mathcal{Q}_{\mathcal{X}}\right) \coloneqq 2 \sup_{A' \in \mathcal{A}} \left|\mathcal{P}_{\mathcal{X}}(A') - \mathcal{Q}_{\mathcal{X}}(A')\right|$$

It obviously is sufficient to consider only those domain subsets were the potential hypotheses predict different outputs $\mathcal{H}\Delta\mathcal{H} \coloneqq \{\{x \in \mathcal{X} \mid h(x) \neq h'(x)\} \mid h, h' \in \mathcal{H}\}$.

Since we select a hypothesis that should perform well with respect to $\mathcal{P}$ based solely on the information from $\mathcal{Q}$, we need to assume that there exists a hypothesis h^* that performs well on both distributions. This hypothesis has been named the low-error joint predictor by Ben-David et al. (2006):

Definition 6. (Low-error joint predictor.) We call a hypothesis h^* a low-error joint predictor, if

$$\mathbb{E}_{\mathcal{P}}\left[\ell(y, h^*(x))\right] + \mathbb{E}_{\mathcal{Q}}\left[\ell(y, h^*(x))\right] \approx \mathbb{E}_{\mathcal{P}}\left[\mathcal{H}\right] + \mathbb{E}_{\mathcal{Q}}\left[\mathcal{H}\right]$$

were $\mathbb{E}_{\mathcal{P}}\left[\mathcal{H}\right] \coloneqq \inf_{h \in \mathcal{H}} \mathbb{E}_{\mathcal{P}}\left[\ell(y, h(x))\right]$ and $\mathbb{E}_{\mathcal{Q}}\left[\mathcal{H}\right] \coloneqq \inf_{h \in \mathcal{H}} \mathbb{E}_{\mathcal{Q}}\left[\ell(y, h(x))\right]$.

It has been shown by Ben-David et al. (2006; 2010b) that the combination of a small distance $\mathrm{d}_{\mathcal{H}\Delta\mathcal{H}}\left(\mathcal{P}_{\mathcal{X}}, \mathcal{Q}_{\mathcal{X}}\right)$ and the existence of h^* is necessary and sufficient for A to be a (ϵ, δ)-learner (making the covariate shift assumption redundant if both of the former conditions hold).

For our first proposed test, we train a model once on the original training data to get an estimate of $\mathbb{E}_{\mathcal{P}}\left[\mathcal{H}\right]$, and also on compositionally synthesized training data to get an estimate for $\mathbb{E}_{\mathcal{P}}\left[\ell(y, h_T(x))\right]$. Calculating the difference between these estimates allows us to estimate an approximate magnitude for ϵ. If this magnitude is small, DA was successful. This allows us to infer a small $\mathcal{A}$-distance between the compositionally synthesized and the original data distributions. Hence, we conclude that the original data generating process must be compositional as well.

If h^* does not exist, we can still test the similarity of P and Q under the covariate shift assumption based on the expected risk ratio. The following theorem takes inspiration from Ben-David & Urner (2014) who observed that a bounded ratio between $\mathcal{P}_{\mathcal{X}}$ and $\mathcal{Q}_{\mathcal{X}}$ implies a bounded ratio of the expected risks calculated with respect to $\mathcal{P}$ and $\mathcal{Q}$ for any learner.

Theorem 1. (Bounded expected risk and density.) Let $\mathcal{P}$ and $\mathcal{Q}$ be two absolutely continuous probability distributions on $\mathcal{X} \times \mathcal{Y}$ for which the covariate shift assumption holds such that $f_{\mathcal{P}}(x, y) = f(y \mid x) f_{\mathcal{P}}(x)$ and $f_{\mathcal{Q}}(x, y) = f(y \mid x) f_{\mathcal{Q}}(x)$ are the corresponding densities. Further let $\mathcal{H}$ be a hypotheses class of learners, $\ell \geq 0$ be a loss function, and $\underline{C}, \overline{C} \in \mathbb{R}^+$. Then,

$$\forall x \in \mathcal{X} \colon \underline{C} f_{\mathcal{Q}}(x) \leq f_{\mathcal{P}}(x) \leq \overline{C} f_{\mathcal{Q}}(x) \iff \forall h \in \mathcal{H} \colon \underline{C}\mathbb{E}_{\mathcal{Q}}\left[\ell\left(x, h(x)\right)\right] \leq \mathbb{E}_{\mathcal{P}}\left[\ell\left(x, h(x)\right)\right] \leq \overline{C}\mathbb{E}_{\mathcal{Q}}\left[\ell\left(x, h(x)\right)\right].$$

Further $\underline{C} \leq 1$ and $\overline{C} \geq 1$.

Proof. To establish sufficiency, let us assume that $\forall x \in \mathcal{X} \colon \underline{C} f_{\mathcal{Q}}(x) \leq f_{\mathcal{P}}(x) \leq \overline{C} f_{\mathcal{Q}}(x)$. Then

$$\forall (x, y) \in \mathcal{X} \times \mathcal{Y} \colon \underline{C} \underbrace{f(y \mid x) f_{\mathcal{Q}}(x)}_{= f_{\mathcal{Q}}(x,y)} \leq \underbrace{f(y \mid x) f_{\mathcal{P}}(x)}_{= f_{\mathcal{P}}(x,y)} \leq \overline{C} \underbrace{f(y \mid x) f_{\mathcal{Q}}(x)}_{= f_{\mathcal{Q}}(x,y)}$$

Because ℓ is non-negative for all h, we can conclude by the monotonicity and linearity of the integral that

$$\forall h \in \mathcal{H} \colon \underline{C}\mathbb{E}_{\mathcal{Q}}\left[\ell\left(x, h(x)\right)\right] \leq \mathbb{E}_{\mathcal{P}}\left[\ell\left(x, h(x)\right)\right] \leq \overline{C}\mathbb{E}_{\mathcal{Q}}\left[\ell\left(x, h(x)\right)\right].$$

To prove necessity, we first assume that $\forall h \in \mathcal{H} \colon \underline{C}\mathbb{E}_{\mathcal{Q}}\left[\ell\left(x, h(x)\right)\right] \leq \mathbb{E}_{\mathcal{P}}\left[\ell\left(x, h(x)\right)\right]$ holds. Let us assume for the moment that $\forall x \in \mathcal{X} \colon \underline{C} f_{\mathcal{Q}}(x) > f_{\mathcal{P}}(x)$. Repeating the argument made to establish sufficiency, the momentary assumption made above implies that $\forall h \in \mathcal{H} \colon \underline{C}\mathbb{E}_{\mathcal{Q}}\left[\ell\left(x, h(x)\right)\right] > \mathbb{E}_{\mathcal{P}}\left[\ell\left(x, h(x)\right)\right]$. Obviously this conclusion contradicts our first assumption. Therefore we have to conclude that $\forall x \in \mathcal{X} \colon \underline{C} f_{\mathcal{Q}}(x) \leq f_{\mathcal{P}}(x)$. Repeating this argument for the second inequality finishes the proof.

To demonstrate that $\underline{C} \leq 1$, we recognize that $\underline{C} f_{\mathcal{Q}}(x) \leq f_{\mathcal{P}}(x) \implies \underline{C} \int f_{\mathcal{Q}}(x) \leq \int f_{\mathcal{P}}(x)$ which directly establishes this fact. The same argument can be repeated to show that $\overline{C} \geq 1$. □

Under the rather unproblematic assumption that $\mathbb{E}_{\mathcal{Q}}\left[\ell\left(x, h(x)\right)\right] > 0$, and by defining $\frac{f_{\mathcal{P}}(x)}{f_{\mathcal{Q}}(x)} = 1$ for points in $\mathcal{X}$ whose densities are zero, we obtain bounds on the ratios of densities and expected risks:

$$\forall x \in \mathcal{X} \colon \underline{C} \leq \frac{f_{\mathcal{P}}(x)}{f_{\mathcal{Q}}(x)} \leq \overline{C} \iff \forall h \in \mathcal{H} \colon \underline{C} \leq \frac{\mathbb{E}_{\mathcal{P}}\left[\ell\left(x, h(x)\right)\right]}{\mathbb{E}_{\mathcal{Q}}\left[\ell\left(x, h(x)\right)\right]} \leq \overline{C}.$$

We utilize this relation to define a second empirically testable criterion. We evaluate a trained learner on the original test set to estimate $\mathbb{E}_{\mathcal{P}}\left[\ell\left(x, h(x)\right)\right]$, and on the compositionally synthesized test set to estimate $\mathbb{E}_{\mathcal{Q}}\left[\ell\left(x, h(x)\right)\right]$. The ratio of these estimates is an estimate of the corresponding expected risk ratio. This allows us to assess whether $\underline{C}$ and $\overline{C}$ are close to one, and by the above result, to infer whether $f_{\mathcal{P}}(x, y)$ and $f_{\mathcal{Q}}(x, y)$ are close. If so, we conclude that the original data generating process is compositional as well.

4 Methods

4.1 Symbolic Representation Learning

The empirical deconstruction process of multivariate time series shown in Figure 2 first needs to identify elementary components that can in a second step be used to synthesize new time series in a compositional manner. Since the composition technique used in the second step relies on discrete representations of sub-segments of time series, we need to transform sub-sequences of real-valued vectors into symbolic representations. The central concept in this context is the notion of a symbol space given in Definition 7.

Definition 7. (Symbol space.) Let $(\mathcal{X}, \mathrm{d})$ be a finite dimensional metric space with metric d and $c_1, \ldots, c_k \in \mathcal{X}$. Then the partition of $\mathcal{X}$ given by:

$$S_i = \left\{ x \in \mathcal{X} \mid c_i = \underset{j=1,\ldots,k}{\arg\min} \left(\mathrm{d}(x, c_j) \right) \right\}$$

for all $i = 1, \ldots, k$ is called the symbol space of $\mathcal{X}$ with symbols S_i and centroids c_i.

In order to transform a time series into a chain of symbols, we break an n-dimensional multivariate time series of length T into consecutive non-overlapping blocks of length Δ subject to $T \mod \Delta = 0$. Next we map these blocks to their corresponding symbol in the symbol space and arrange them in the same order as the blocks. The domain of this mapping can be either the original input space $\mathcal{M}_{n,\Delta}$, or the space of the learned neural block representations. In the first case, a straightforward application of traditional symbolic dynamics methods (Lind & Marcus, 1995; Williams, 2004) would require an alphabet size that grows exponentially with the number of input dimensions. This is infeasible even for multivariate time series of around 100 clinical variables. An approach to circumvent this problem is to randomly select a feasible number of centroids in the input domain $\mathcal{M}_{n,\Delta}$.

In the second case, we exploit the representations learned by a neural network and apply k-means[3] clustering to create a symbolic representation of the time series with a computational learning cost that is linear in the embedding dimension. The learned representations consist of the hidden states of the encoder of the TSF Transformer described in Section 5.1.2. This Transformer model was trained to predict three future hours based on the current three hours on the training set, with hyperparameter settings as described in Appendix A.1 (except the hidden size set to 50).

4.2 Data-Driven Compositional Data Synthesization

Andreas (2020) presented a linguistically motivated data-driven approach to induce compositional structure in symbol sequences in NLP. This algorithm can be readily applied to induce compositional structure from symbolic representations of time series. The key inspiration of the algorithm is the *distributional structure* of language (Harris, 1954) according to which "You shall know a word by the company it keeps!" (Firth, 1957). This principle is grounded in the fact that words that are used and occur in the same contexts tend to purport similar meanings. The method of Andreas (2020) exploits the distributional principle to generate novel valid sequences by exchanging subsequences with similar meanings into new contexts.[4] An illustrative example of the distributional structure of language and how the method uses it is given in Figure 3.

---

[3] We used the scikit-learn (Pedregosa et al., 2011) version of k-means clustering, implementing Lloyd's standard algorithm (Lloyd, 1982) by default.

[4] Note that this algorithm works on historical data and is but one possible approximation of the original data generation process assumed to underlie (clinical) time series. We consider its purely data-driven nature and its impartiality towards the underlying compositional process an advantage for the current exposition.

training data	M	$\boldsymbol{A}$	N	$\boldsymbol{B}$
	She	***picks***	*the suitcase*	***up***
	M	$\boldsymbol{C}$	N	$\boldsymbol{D}$
	She	***puts***	*the suitcase*	***down***
	X	$\boldsymbol{A}$	Y	$\boldsymbol{B}$
	He	***picks***	*the box*	***up***
augmentation	X	$\boldsymbol{C}$	Y	$\boldsymbol{D}$
	He	***puts***	*the box*	***down***

- Identify interchangeable *fragments* as (discontinuous) symbol sequences that appear in similar contexts (*environments*) in the training data, for example, $\boldsymbol{A}\dots\boldsymbol{B}$ and $\boldsymbol{C}\dots\boldsymbol{D}$ in context $M\dots N$.
- Find additional contexts in which interchangeable fragments appear, forming a *template* for substituting in another fragment, for example, context $X\dots Y$ for fragment $\boldsymbol{A}\dots\boldsymbol{B}$.
- Augment the data with a synthesized training example by exchanging fragments in a template, for example, substitute $\boldsymbol{C}\dots\boldsymbol{D}$ for $\boldsymbol{A}\dots\boldsymbol{B}$ in context $X\dots Y$.

Figure 3: Compositional data synthesization procedure for symbol sequences adapted from Andreas (2020). Symbol sequences $A\dots B$, $C\dots D$, $M\dots N$, and $X\dots Y$ are generic and can be discontinuous. An illustration with with natural language sentences if given in the respective second lines.

In the following, we will present a formal account of the compositional data synthesization method illustrated in Figure 3. We will introduce the notions of fragment, template, and environment of a sequence, and give a formal definition of the synthesization process and provide pseudo-code for the algorithm used in our work (see Algorithm 1 in Appendix A.3).

Definition 8. (Sequence.) A sequence of length k is a k-tuple whose elements are called symbols.

Definition 9. (Fragment.) Given a sequence s and a set of indices $I_{\text{frg}} \in 2^{\{1,\dots,k\}}$. A fragment is the tuple of subsequences of s given by consecutive indices in I_{frg}.

Definition 10. (Template.) Given a sequence s with length k and a pair of index sets $I_{\text{frg}} \in 2^{\{1,\dots,k\}}$ and $I_{\text{tpl}} \coloneqq \{1,\dots,k\} \setminus I_{\text{frg}}$, a template is defined as the tuple whose symbols are identical to those of s for all the indices of I_{tpl}, and a slot symbol $\bigstar$ for all indices in I_{frg} where consecutive slot symbols are reduced to one.

Definition 11. (Environment.) Given a template $t = (t_1,\dots,t_k)$ and a window size w. Let $I_{\text{env}} \coloneqq \big\{ i \in \{1,\dots,k\} \mid \bigstar \in (t_{\max(1,i-w)},\dots,t_{\min(k,i+w)}) \big\}$. The corresponding subsequence of t given by I_{env} is called the environment of t (with respect to the window size w).

Definition 12. (Insert). Given a template $t = (t_1,\dots,t_k)$ and a fragment $f = (f_1,\dots,f_m)$ where $\operatorname{card}(f) = \operatorname{card}(t_i \in t \mid t_i = \bigstar)$, and let $i_1,\dots,i_m$ be the slot indices of t, then:

$$\begin{aligned} \text{insert}\colon\ & (t_1,\dots,t_k)\times(f_1,\dots,f_m)\to(s_1,\dots,s_n)\\ & (t_1,\dots,t_k)\times(f_1,\dots,f_m)\mapsto \text{flatten}\left(\left(s_i=\begin{cases} f_j, & i=i_j\\ t_i, & t_i\neq\bigstar\end{cases}\right)_{i=1}^{k}\right)\end{aligned}$$

combines t and f by replacing the slot symbols of t with the symbols of the corresponding subsequences of f to a sequence s.

Definition 13. (Compositional data synthesization.) Let there be a fragment f and two non-identical sequences s_a and s_c such that $s_a = \text{insert}(t_a, f)$ and $s_c = \text{insert}(t_c, f)$. Furthermore, let there be a sequence $s_b = \text{insert}(t_b, f_b)$ such that $f_b \neq f$ and $\text{environment}(t_a, w) = \text{environment}(t_b, w)$. Then $s_{\text{syn}} \coloneqq \text{insert}(t_c, f_b)$ is a compositionally created synthetic symbol sequence that is different to s_a, s_b and s_c.

Table 1: Number of patient stays in clinical time series.

Data split	**MIMIC-III**	**eICU**
train	12,304	49,730
dev	3,169	12,433
test	3,803	3,008

In order to translate a sequence $s = \text{insert}(t, f)$ into a time series, we choose a time series whose symbolization is identical to the sequence s. Because symbolization is index-preserving, we can directly map time series sections to t and f by simply replacing their symbols with the time series blocks of the same index. To compose a synthetic time series $x_{\text{syn}} \coloneqq \text{insert}(t_c, f_b)$ we obtain time series x_b and x_c with the corresponding symbolizations s_b and s_c and desymbolize t_c and f_b accordingly.

5 Experiments

5.1 Experimental Setup

5.1.1 Data

In our experiments, we focus on two clinical time series datasets: MIMIC-III (Johnson et al., 2016) and eICU (Pollard et al., 2018). Both databases contain anonymized information from ICU patients, including physiological measurements (e.g., heart rate, blood pressure), medications (e.g., dobutamin, epinephrine), interventions (e.g., intubation), lab test results (e.g., blood cultures,) and clinical notes.

We extracted features that were frequently tracked during a patient's stay, namely 131 features for MIMIC-III, and 98 for eICU (a complete list can be found in Appendix A.2). For each patient stay we extracted the first 48 hours, but only took stays into account that had at least one feature measured per hour. Furthermore, we removed patients that did not stay long enough in the ICU. The remaining clinical variables were standardized, and the resulting datasets were partitioned into training, validation and test data (see Table 1 for exact numbers).

Notably, our data can be characterized as sparse and irregularly sampled (Tipirneni & Reddy, 2022; Horn et al., 2020). This is because some features like heart rate or blood pressure are recorded every 5 minutes, other features like those only available through blood sampling, are available only every 24 hours. We store these sparse multivariate time series in a database of n quadruplets $\mathrm{S} = \{(f_i, t_i, v_i, n_i)\}_{i=1}^{n}$, where $f_i \in F$ is a clinical variable identifier, $t_i \in \mathbb{R}_{\geq 0}$ is a time index, $v_i \in \mathbb{R}$ the observed value of f_i at t_i, and n_i the unique stay identifier. In our experiments we use a dense representation of the data, where every timestep is a vector of feature values representing one hour. We construct this vector by choosing the first observed value during the represented hour for each feature. If no value was observed, we impute zero which corresponds to the mean value due to standardization of the data. Additionally, a mask indicating whether a value was imputed is generated and appended to the vector. Based on this representation, we define the sparsity of a feature as the relative frequency of imputations.

To generate a synthetic dataset, we first need to assign symbols to the training data, as described in Section 4.1, by either choosing random centroids in the input space (input), or by applying k-means clustering to the learned representations (embd), with a variable number of centroids (#Syms). For the conducted experiments, we represent 3 hours with one symbol and then apply the compositional data synthesization algorithm described in Section 4.2 to generate compositional synthetic data versions, or use the CutMix algorithm (Yun et al., 2019) directly on the time series to produce non-compositional synthetic data.

5.1.2 Time Series Forecasting with Transformers

In our experiments, we use a Transformer model with an autoregressive decoder that generates an output vector $\hat{y}_t \in \mathbb{R}^{|F|}$ (where $|F|$ is the number of features used). The predicted output $\hat{y}_t$ is a function of the

Table 2: Estimates of $\hat{\epsilon} \approx \mathbb{E}_{\text{orig}}\left[\ell(y, h_T(x))\right] - \mathbb{E}_{\text{orig}}\left[\mathcal{H}\right]$ where T was generated synthetically for MIMIC-III and eICU based on all models obtained during training.

Synthetic Dist	**Symbolization**	**MIMIC-III**	**eICU**
		$\hat{\epsilon}$ (SE)	$\hat{\epsilon}$ (SE)
CutMix		0.753 (0.015)	0.911 (0.012)
CDS	input	0.070 (0.015)	0.064 (0.012)
CDS	embd	0.110 (0.019)	0.055 (0.015)

Table 3: Estimate of $\mathbb{E}_{\text{syn}}\left[\ell\left(x, h^*(x)\right)\right] / \mathbb{E}_{\text{orig}}\left[\ell\left(x, h^*(x)\right)\right]$ where h^* is the best model obtained by training on the original training data.

Synthetic Dist	**Symbolization**	**MIMIC-III**		**eICU**	
		MSE*	Ratio	MSE*	Ratio
CutMix		10.201	1.375	7.648	1.445
CDS	input	7.398	0.997	5.279	0.998
CDS	embd	7.230	0.974	5.086	0.961

history $\hat{y}_{<t}$ of predicted timesteps until time t, the encoded input x, and the model parameters θ:

$$\hat{y}_t = f_\theta(\hat{y}_{<t}, x) \tag{2}$$

To perform long-term TSF using the autoregressive setup, the outputs $\hat{y}_t$ from each time step $t = 1, \ldots, T$ are concatenated.

We employ a standard transformer architecture (Vaswani et al., 2017) as our model of choice, the hyperparameters of which can be found in Appendix A.1. The encoder takes as input the first 24 hours of a timeseries in a densified format (Section 5.1.1). The decoder then generates the next 24 hours of the timeseries. The complete model is trained with masked mean squared error (MSE) (see Appendix A.4). The model can either be trained on the original data or the various versions of synthetic data. A model that has been trained on synthetic data can be further finetuned with original data.

5.2 Experimental Results

In the following, we present an analysis of the compositionality of the data generation process underlying original clinical time series. This is conducted by analyzing the two empirically testable criteria developed in Section 3.3 (see Section 5.2.1 for the first design, and Section 5.2.2 for the second design).

5.2.1 Test 1: Utility of Synthesized Data for TSF Training

The first empirically testable criterion described in Section 3.3 infers the compositionality of the original data generating process from the estimated expected risk difference $\mathbb{E}_{\text{orig}}\left[\ell(y, h_T(x))\right] - \mathbb{E}_{\text{orig}}\left[\mathcal{H}\right]$. $\mathbb{E}_{\text{orig}}\left[\mathcal{H}\right]$ is estimated by training a TSF model on the original time series data and evaluate it on the original test data. $\mathbb{E}_{\text{orig}}\left[\ell(y, h_T(x))\right]$ is estimated by training a TSF model on synthetic data and evaluate it on the original test data. The synthetic data can be generated either non-compositional (CutMix) or compositional (CDS) with symbolization obtained by clustering in input or neural embedding space. The estimate of $\hat{\epsilon}$ and its standard error is obtained by training a linear mixed effects model (Demidenko, 2013; Bates et al., 2015) on the evaluation scores of models obtained from three different random seeds for optimization, and another three random seeds for symbolization. The main result is given in Table 2: It shows values of $\hat{\epsilon}$ that are close to zero for CDS training, and an order of magnitude larger for training on randomization-based synthetic data, on both datasets of clinical time series.

5.2.2 Test 2: Utility of Synthesized Data as TSF Test Data

Table 3 presents the experimental results for the second empirically testable criterion established in Section 3.3. This test is based on a model h^* that is obtained by training on original time series data and its evaluation on original and synthetic test data. The goal is to infer a uniform bound for the ratio of the synthetic and original data distribution based on estimates of the corresponding expected risks $\mathbb{E}_{\text{syn}}\left[\ell\left(x, h^*(x)\right)\right]$ and $\mathbb{E}_{\text{orig}}\left[\ell\left(x, h^*(x)\right)\right]$.

We see that the obtained ratios are close to 1 for evaluations on original data and compositionally synthesized data, but not for test data synthesized via the randomization-based CutMix method. This again supports our reasoning of compositionality of the data generation process underlying the original time series data.

5.2.3 Further Experimental Evaluation

In addition to this quantitative evaluation founded in domain adaptation theory, we analyze the distributional properties of the compositionally created data with respect to the original data (see Appendix A.6). We find that the Hellinger distance of distributions of original and compositionally synthesized data show that the data are close in unigram space — indicating similar distributions of symbols — but are further apart in higher n-gram space — indicated dissimilar distributions of compounds. These distributional properties are desired for benchmark data for compositional generalization (Keysers et al., 2020), and are recreated by our compositional data synthesization method.

Lastly, we present an interpretation of the symbolic representations used in our pipeline (see Appendix A.7). This qualitative assessment shows that meaningful physiological states can be learned by symbolic representation learning.

6 Conclusion

We presented an investigation of the question whether non-linguistic time series — here time series of clinical measurements — also exhibit the intriguing property of compositionality of natural language sequences, namely the characteristics of being generated by a process that combines elementary components following a compositional rule system. If so, we could analyze clinical time series as a series of subsequences that are emitted by an ordered sequence of physiological states, and create synthetic data for notoriously sparse and low-resource data situations in clinical TSF. We showed that a method from NLP that is based entirely on distributional properties of data allows us to synthesize novel combinations of elementary time series subsequences with most interesting properties: Training and testing of TSF models on compositionally synthesized time series yields a similar utility than training and testing on original time series data, allowing us to conclude that the data generation process underlying the original time series data can in fact be characterized as compositional.

Our experiments are based on applying a pipeline of data-driven symbolic representation learning and compositional data augmentation to clinical time series data, and we find consistent improvements over randomization-based data synthesization on two clinical time series datasets. We show that our empirical tests can be motivated in domain adaptation theory, drawing on a possible inference about the distributions of the data generation process of source and target data from the performance of training and testing models on these data. This theoretical motivation allows us to make broader claims on the compositionality of time series data beyond clinical time series, opening the doors to further research on compositional data augmentation and domain adaptation in general time series modeling tasks.

A limitation of our work is the lack of an embedding into a specific clinical problem, based on problem-specific clinical measurements, and an evidence-based clinical interpretation. This will be an important task for future work.

References

Ekin Akyürek, Afra Feyza Akyürek, and Jacob Andreas. Learning to recombine and resample data for compositional generalization. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=PS3IMnScugk`.

Jacob Andreas. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, Online, 2020. URL `10.18653/v1/2020.acl-main.676`.

Douglas Bates, Martin Mächler, Benjamin M. Bolker, and Steven C. Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015. URL `https://doi.org/10.18637/jss.v067.i01`.

Shai Ben-David and Ruth Urner. Domain adaptation - can quantity compensate for quality? *Annals of Mathematics and Artificial Intelligence*, 70:185–202, 2014. URL `https://doi.org/10.1007/s10472-013-9371-9`.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In B. Schölkopf, J. Platt, and T. Hoffman (eds.), *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. URL `https://proceedings.neurips.cc/paper_files/paper/2006/file/b1b0432ceafb0ce714426e9114852ac7-Paper.pdf`.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010a. URL `https://link.springer.com/article/10.1007/s10994-009-5152-4`.

Shai Ben-David, Tyler Lu, Teresa Luu, and David Pal. Impossibility theorems for domain adaptation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 129–136, Chia Laguna Resort, Sardinia, Italy, 2010b. URL `https://proceedings.mlr.press/v9/david10a.html`.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013. URL `https://doi.org/10.1109/TPAMI.2013.50`.

Chengtai Cao, Fan Zhou, Yurou Dai, Jianping Wang, and Kunpeng Zhang. A survey of mix-based data augmentation: Taxonomy, methods, applications, and explainability. *ACM Comput. Surv.*, 57(2), October 2024. ISSN 0360-0300. doi: 10.1145/3696206. URL `https://doi.org/10.1145/3696206`.

Noam Chomsky. *Syntactic Structures.* De Gruyter Mouton, 1957.

Noam Chomsky. On certain formal properties of grammars. *Information and Control*, 2(2):137–167, 1959. URL `https://doi.org/10.1016/S0019-9958(59)90362-6`.

Eugene Demidenko. *Mixed Models: Theory and Applications with R.* Wiley, 2013. URL `https://doi.org/10.1002/0471728438`.

John R. Firth. A synopsis of linguistic theory, 1930-55. In *Studies in Linguistic Analysis*, pp. 1–31. Blackwell, 1957.

Hamid Ghaderi, Brandon Foreman, Amin Nayebi, Sindhu Tipirneni, Chandan K. Reddy, and Vignesh Subbian. A self-supervised learning-based approach to clustering multivariate time-series data with missing values (SLAC-Time): An application to TBI phenotyping. *Journal of Biomedical Informatics*, 143:104401, 2023. URL `https://doi.org/10.1016/j.jbi.2023.104401`.

Peikun Guo, Huiyuan Yang, and Akane Sano. Empirical study of mix-based data augmentation methods in physiological time series data. In *The 11th IEEE International Conference on Healthcare Informatics (IEEE ICHI)*, Houston, TX, USA, 2023. URL `https://doi.org/10.48550/arXiv.2309.09970`.

Zellig S. Harris. Distributional structure. *WORD*, 10(2-3):146–162, 1954. URL `https://doi.org/10.1080/00437956.1954.11659520`.

I. Higgins, Loïc Matthey, A. Pal, C. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France, 2017. URL `https://openreview.net/forum?id=Sy2fzU9gl`.

Max Horn, Michael Moor, Christian Bock, Bastian Rieck, and Karsten Borgwardt. Set functions for time series. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, Online, 2020. URL `https://proceedings.mlr.press/v119/horn20a.html`.

Chen Huang, Peixin Qin, Wenqiang Lei, and Jiancheng Lv. Towards equipping transformer with the ability of systematic compositionality. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. URL `10.1609/aaai.v38i16.29788`.

Theo M.V. Janssen. Compositionality: its historic context. In *The Oxford handbook of compositionality*, pp. 19–46. Oxford University Press, 2012. URL `https://eprints.illc.uva.nl/1239/2/LP-1996-03.text.pdf`.

Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Li wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1):160035, 2016. URL `https://doi.org/10.1038/sdata.2016.35`.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations (ICLR)*, virtual, 2020. URL `https://openreview.net/forum?id=SygcCnNKwr`.

Najoung Kim and Tal Linzen. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, 2020. URL `10.18653/v1/2020.emnlp-main.731`.

Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Larning (ICML)*, Stockholm, Sweden, 2018. URL `http://proceedings.mlr.press/v80/lake18a/lake18a.pdf`.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253, 2017. URL `https://doi.org/10.1017/S0140525X16001837`.

Baihan Lin, Djallel Bouneffouf, and Irina Rish. A survey on compositional generalization in applications. *arXiv*, abs/2302.01067, 2023. URL `https://doi.org/10.48550/arXiv.2302.01067`.

Douglas Lind and Brian Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, 1995.

Stuart P. Lloyd. Least square quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. URL `https://doi.org/10.1109/TIT.1982.1056489`.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, Long Beach, California, USA, 2019. URL `http://proceedings.mlr.press/v97/locatello19a.html`.

Qianli Ma, Jiawei Zheng, Sen Li, and Gary W Cottrell. Learning representations for time series clustering. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. URL `https://proceedings.neurips.cc/paper_files/paper/2019/file/1359aa933b48b754a2f54adb688bfa77-Paper.pdf`.

Richard Montague. Universal grammar. *Theoria*, 36:373–398, 1970.

Milton Montero, Jeffrey Bowers, Rui Ponte Costa, Casimir Ludwig, and Gaurav Malhotra. Lost in latent space: Examining failures of disentangled models at combinatorial generalisation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/41ca8a0eb2bc4927a499b910934b9b81-Paper-Conference.pdf`.

Milton Llera Montero, Casimir JH Ludwig, Rui Ponte Costa, Gaurav Malhotra, and Jeffrey Bowers. The role of disentanglement in generalisation. In *International Conference on Learning Representations (ICLR)*, 2021. URL `https://openreview.net/forum?id=qbH974jKUVy`.

Barbara H. Partee. Compositionality. In *Varieties of formal semantics*, volume 3, pp. 281–311. 1984.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research (JMLR)*, 12:2825–2830, 2011. URL `https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf`.

Tom J. Pollard, Alistair E. W. Johnson, Jesse D. Raffa, Leo A. Celi, Roger G. Mark, and Omar Badawi. The eICU collaborative research database, a freely available multi-center database for critical care research. *Scientific Data*, 5(180178), 2018. URL `https://doi.org/10.1038/sdata.2018.178`.

Linlu Qiu, Peter Shaw, Panupong Pasupat, Pawel Nowak, Tal Linzen, Fei Sha, and Kristina Toutanova. Improving compositional generalization with latent structure and data augmentation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Seattle, United States, 2022. URL `10.18653/v1/2022.naacl-main.323`.

Jacob Russin, Jason Jo, Randall O'Reilly, and Yoshua Bengio. Compositional generalization by factorizing alignment and translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, Online, 2020. URL `10.18653/v1/2020.acl-srw.42`.

Laurent Sartran, Samuel Barrett, Adhiguna Kuncoro, Miloš Stanojević, Phil Blunsom, and Chris Dyer. Transformer grammars: Augmenting transformer language models with syntactic inductive biases at scale. *Transactions of the Association for Computational Linguistics*, 10:1423–1439, 2022. URL `10.1162/tacl_a_00526`.

Lukas Schott, Julius Von Kügelgen, Frederik Träuble, Peter Vincent Gehler, Chris Russell, Matthias Bethge, Bernhard Schölkopf, Francesco Locatello, and Wieland Brendel. Visual representation learning does not generalize strongly within the same domain. In *International Conference on Learning Representations (ICLR)*, 2022. URL `https://openreview.net/forum?id=9RUHPlladgh`.

Zoltán Gendler Szabó. Compositionality. *Stanford Encylopedia of Philosophy*, 2020. URL `https://plato.stanford.edu/entries/compositionality/`.

Sindhu Tipirneni and Chandan K. Reddy. Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series. *ACM Transactions on Knowledge Discovery from Data*, 16(6), 2022. doi: 10.1145/3516367. URL `https://doi.org/10.1145/3516367`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, Long Beach, CA, 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

Thaddäus Wiedemer, Prasanna Mayilvahanan, Matthias Bethge, and Wieland Brendel. Compositional generalization from first principles. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, USA, 2023. URL `https://openreview.net/forum?id=LqOQ1uJmSx`.

Susan Williams. Introduction to symbolic dynamics. In *Proceedings of the Symposia in Applied Mathematics*, volume 60, 1-12 2004.

Zhenlin Xu, Marc Niethammer, and Colin Raffel. Compositional generalization in unsupervised compositional representation learning: a study on disentanglement and emergent language. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, USA, 2022. URL `https://papers.neurips.cc/paper_files/paper/2022/file/9f9ecbf4062842df17ec3f4ea3ad7f54-Paper-Conference.pdf`.

Hong Yang and Travis Desell. Robust augmentation for multivariate time series classification. *arXiv*, abs/2201.11739, 2022. URL `https://doi.org/10.48550/arXiv.2201.11739`.

Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of ICCV*, Seoul, Korea (South), 2019. URL `10.1109/ICCV.2019.00612`.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018. URL `https://openreview.net/forum?id=r1Ddp1-Rb`.

A Appendix

A.1 Hyperparameters

Table 4: Hyperparameter settings for training. Best settings chosen on development data are shown in bold face.

Parameter	MIMIC-III	eICU
Embedding Size	128, 256, **512**	**256**, 512, 1024
Hidden Size Encoder	128, 256, **512**	**256**, 512, 1024
Hidden Size DMS Decoder	128, 256, **512**	**256**, 512, 1024
Hidden Size IMS Decoder	Output Dimensionality	Output Dimensionality
# Encoder Layers	1, **2**	1, **2**, 3
# Decoder Layers	**1**, 2	**1**
Learning Rate	**0.0005**	**0.0005**
Finetune Learning Rate	**0.0001**	**0.0001**
Batch Size	32	32
Attention Heads Encoder	2, 4, **8**	**8**
Attention Heads Decoder	**1**, 2, 4	**1**,2,4
Dropout	**0.05**, 0.1, 0.2	**0.05**
Epochs	100	600
Patience	6	6
Random Seed	Unixtime variation	Unixtime variation

A.2 Feature Lists

Table 5: Feature list for MIMIC-III: Besides the following 131 dynamic variables, only age and gender were extracted. The 15 variables marked with an asterisk are directly used for calculating the SOFA score.

ALP
ALT
AST
Albumin
Albumin 25%
Albumin 5%
Amiodarone
Anion Gap
BUN
Base Excess
Basophils
Bicarbonate
Bilirubin (Direct)
Bilirubin (Indirect)
Bilirubin (Total)*
CRR
Calcium Free
Calcium Gluconate
Calcium Total
Cefazolin
Chest Tube
Chloride
Colloid
Creatinine Blood*
Creatinine Urine
D5W
DBP*
Dextrose Other
Dobutamine*
Dopamine*
EBL
Emesis
Eoisinophils
Epinephrine*
Famotidine
Fentanyl
FiO2*
Fiber
Free Water
Fresh Frozen Plasma
Furosemide
GCS_eye*
GCS_motor*
GCS_verbal*
GT Flush
Gastric
Gastric Meds
Glucose (Blood)
Glucose (Serum)
Glucose (Whole Blood)
HR
Half Normal Saline
Hct
Heparin
Hgb
Hydralazine
Hydromorphone
INR
Insulin Humalog
Insulin NPH
Insulin Regular
Insulin largine
Intubated
Jackson-Pratt
KCl
KCl (Bolus)
LDH
Lactate
Lactated Ringers
Levofloxacin
Lorazepam
Lymphocytes
Lymphocytes (Absolute)
MBP
MCH
MCHC
MCV
Magnesium
Magnesium Sulfate (Bolus)
Magnesium Sulphate
Mechanically ventilated
Metoprolol
Midazolam
Milrinone
Monocytes
Morphine Sulfate
Neosynephrine
Neutrophils
Nitroglycerine
Nitroprusside
Norepinephrine*
Normal Saline
O2 Saturation
OR/PACU Crystalloid
PCO2
PO intake
PO2*
PT
PTT
Packed RBC
Pantoprazole
Phosphate
Piggyback
Piperacillin
Platelet Count*
Potassium
Pre-admission Intake
Pre-admission Output
Propofol
RBC
RDW
RR
Residual
SBP*
SG Urine
Sodium
Solution
Sterile Water
Stool
TPN
Temperature
Total CO2
Ultrafiltrate
Urine*
Vancomycin
Vasopressin
WBC
Weight
pH Blood
pH Urine

Table 6: Feature list for eICU: Besides the following 98 dynamic variables, there are 17 static variables covering age, gender, admission information, and ICU type. The 15 variables marked with an asterisk are directly used for calculating the SOFA score. On the right column, there are 35 drug-related variables. Some of them seem redundant due to different hospitals but can not be merged because of different or not standardized concentrations.

ALP	Lactate	Amiodarone
ALT	Lymphocytes	Dobutamine dose
AST	MBP	Dobutamine ratio*
Albumin	MCH	Dopamine dose
Anion Gap	MCHC	Dopamine ratio*
BUN	MCV	Epinephrine dose
Base Deficit	MPV	Epinephrine ratio*
Base Excess	Magnesium	Fentanyl 1
Basophils	Monocytes	Fentanyl 2
Bedside Glucose	Neutrophils	Fentanyl 3
Bicarbonate	O2 L/%	Furosemide
Bilirubin (Direct)	O2 Saturation	Heparin 1
Bilirubin (Total)*	PT	Heparin 2
Bodyweight (kg)	PTT	Heparin 3
CO2 (Total)	PaCO2	Heparin vol
Calcium	PaO2*	Insulin 1
Chloride	Phosphate	Insulin 2
Creatinine (Blood)*	Platelets*	Insulin 3
Creatinine (Urine)	Potassium	Midazolam 1
DBP*	Protein (Total)	Midazolam 2
Eoisinophils	RBC	Milrinone 1
EtCO2	RDW	Milrinone 2
FiO2*	RR	Nitroglycerin 1
Fibrinogen	SBP*	Nitroglycerin 2
GCS eye*	Sodium	Nitroprusside
GCS motor*	Stool	Norepinephrine 1
GCS verbal*	Temperature	Norepinephrine 2
Glucose	Troponin - I	Norepinephrine ratio*
HR	Urine*	Pantoprazole
Hct	WBC	Propofol 1
Hgb	pH	Propofol 2
INR		Propofol 3
		Vasopressin 1
		Vasopressin 2
		Vasopressin 3

A.3 Compositional data synthesization algorithm

Algorithm 1 Compositional data synthesization (CDS) algorithm.

```
frg_to_tpl = defaultdict(list)
tpl_to_frg = defaultdict(list)
env_to_tpl = defaultdict(list)
for seq in dataset:
    #frg, tpl and env are index sets
    for frg in frgs(seq):
        tpl = template(seq, frg)
        env = environment(tpl, window_size)
        #fetch symbols
        frg_s = get_syms(seq, frg, value_type='frg')
        tpl_s = get_syms(seq, tpl, value_type='tpl')
        env_s = get_syms(seq, env, value_type='env')
        #append maps
        frg_to_tpl[frg_s].append(tpl_s)
        tpl_to_frg[tpl_s].append(frg_s)
        env_to_tpl[env_s].append(tpl_s)

frg_list = list(frg_to_tpl)
while True:
    shuffle(frg_list)
    for frg in frg_list:
        tpl_c_list = list(frg_to_tpl[frg])
        shuffle(tpl_c_list)
        for tpl_c in tpl_c_list:
            #get all tpl for frg without tpl_c
            tpl_a_list = [tpl for tpl in tpl_c_list if tpl != tpl_c]
            shuffle(tpl_a_list)
            for tpl_a in tpl_a_list:
                #retrieve templates with same environment as tpl_a
                for tpl_b in env_to_tpl[environment(tpl_a, window_size)]:
                    #retrieve all fragments for tpl_b
                    for frg_b in tpl_to_frg[tpl_b]:
                        if frg_b != frg:
                        ts_tpl_c, ts_frg_c = get_ts_segments(tpl_c, frg)
                        ts_tpl_b, ts_frg_b = get_ts_segments(tpl_b, frg_b)
                        yield insert(ts_tpl_c, ts_frg_b)
```

A.4 Evaluation Metrics

Given N time series in our dataset, with a prediction window of T hours for TSF, the masked mean squared error (MSE) over hourly prediction vectors $\hat{y}_t^n$ is defined as follows:

$$\text{MSE} = \frac{1}{NT} \sum_{n=1}^{N} \sum_{t=1}^{T} ||(y_t^n - \hat{y}_t^n) \odot m_t^n||_2^2 \tag{3}$$

where $m_t^n \in \{0,1\}^{|F|}$ is a mask indicating if the variables in y_t^n were observed or not, and $\odot$ is a component-wise product. In our experiments, T is set to 24 hours.

For each synthetically generated dataset and for the original data, we train three differently seeded models. In our experiments, we report the average MSE of the three training runs ($\overline{\text{MSE}}$) and the corresponding standard deviation (SD). In addition, we report the best performing model (MSE^*) and the 95% confidence interval for the estimation of the evaluation score of MSE^* on the test set. This is calculated from the sample mean $\overline{\mu}$ and standard deviation $\overline{s}$ of MSE^* scores on a test set of size N as

$$\text{KI}_{.95} = \overline{\mu} - 1.96\frac{\overline{s}}{\sqrt{N}}; \overline{\mu} + 1.96\frac{\overline{s}}{\sqrt{N}}$$

.

A.5 Further Experimental Evaluation of Utility of Synthesized Data: Pretraining versus Finetuning

Tables 7 and 8 present the raw data underlying the analysis given in Table 2. The performance results of various TSF models with respect the above described test are shown in the two columns labeled "Pretraining". The first rows in both tables show evaluation scores as defined in Section A.4 for evaluating a TSF model pretrained on original time series on original test data from MIMIC-III and eICU, respectively. The second row shows evaluation results on the same original test data for a model pretrained on data generated by the CutMix (Yun et al., 2019) method. Similar to Guo et al. (2023); Yang & Desell (2022), we apply this method to randomly selected pairs of time series that are cut at random points where the time series subsegments are exchanged. This technique produces significantly worse evaluation results than a model pretrained on original data. Compared to this, models pretrained on compositionally synthesized data (rows 3-8) perform very close to models trained on original data, with best results for symbolization in input space for MIMIC-III (row 5 in Table 7) and best results for symbolization in embedding space for eICU (row 8 in Table 8). Furthermore, we see that increasing the number of symbols is beneficial. In sum, the closeness of performance of models pretrained on compositionally synthesized data and original data supports our reasoning of compositionality of the data generation process underlying the original time series data.

The last two columns in Tables 7 and 8, labeled "Finetuning on original", show results for a control experiment that performs continued training on original training data for all models, initialized at the weights obtained by pretraining on the data listed in the first column. We see that the models pretrained on CutMix data gain substantially in performance, whereas the models pretrained on compositionally synthesized data only improve by a small margin. This is expected given the inferred distributional similarity of original and compositionally synthesized data.

Overall, the combination of pretraining on compositionally synthesized data and finetuning on original data yields evaluation results that outperform training on original data alone. This is a nice side-effect, showing the quality of compositionally synthesized data for data augmentation in small resource scenarios.

Table 7: Experimental results of training data utility for TSF on MIMIC-III. The first column lists pretraining data (original, randomized synthesization by CutMix, compositional data synthesization (CDS)). The second column specifies the symbolization method (clustering in input space or of neural embeddings). The third column lists the number of clusters/symbols (40, 80, 160). Evaluation results according to the metrics described in Section A.4 are given in the fourth to sixth column. Best results for pretraining on synthesized data and finetuning on original data are highlighted in **bold face**.

Data	**Symbolization**		**Pretraining**		**Finetuning on original**	
	Domain	#Syms	$\overline{\text{MSE}}$ (SD)	MSE^{*} [95KI]	$\overline{\text{MSE}}^{3}$(SD)	MSE^{*} [95KI]
original			7.583^{1}(0.135)	$7.420_{[7.411,7.429]}$		
CutMix			8.336^{2}(0.130)	$8.238_{[8.229,8.247]}$	7.704 (0.223)	$7.561_{[7.553,7.569]}$
CDS	input	40	7.989^{2}(0.234)	$7.757_{[7.748,7.766]}$	7.825 (0.298)	$7.484_{[7.476,7.493]}$
CDS	input	80	7.700^{2}(0.065)	$7.607_{[7.598,7.616]}$	7.585 (0.119)	$7.438_{[7.429,7.447]}$
CDS	input	160	$\mathbf{7.653}^{2}$(0.056)	$\mathbf{7.559}_{[7.550,7.568]}$	7.550 (0.069)	$7.424_{[7.415,7.433]}$
CDS	embd	40	7.846^{1}(0.025)	$7.811_{[7.801,7.820]}$	7.581 (0.088)	$7.492_{[7.482,7.501]}$
CDS	embd	80	7.711^{1}(0.033)	$7.674_{[7.665,7.684]}$	7.556 (0.095)	$7.439_{[7.431,7.448]}$
CDS	embd	160	7.693^{1}(0.041)	$7.637_{[7.627,7.646]}$	**7.455** (0.067)	$\mathbf{7.388}_{[7.379,7.398]}$

[1] N=3 (three seeds for optimization)

[2] N=9 (three seeds for optimization and three for symbolization)

[3] N is the same as for training.

Table 8: Experimental results of training data utility for TSF on eICU. The first column lists pretraining data (original, randomized synthesization by CutMix, compositional data synthesization (CDS)). The second column specifies the symbolization method (clustering in input space or of neural embeddings). The third column lists the number of clusters/symbols (40, 80, 160). Evaluation results according to the metrics described in Section A.4 are given in the fourth to sixth column. Best results for pretraining on synthesized data and finetuning on original data are highlighted in **bold face**.

Data	**Symbolization**		**Pretraining**		**Finetuning on original**	
	Domain	#Syms	$\overline{\text{MSE}}$ (SD)	MSE^* [95KI]	$\overline{\text{MSE}}$[3](SD)	MSE^* [95KI]
original			5.299[1](0.007)	$5.291_{[5.286,5.295]}$		
CutMix			6.210[2](0.045)	$6.144_{[6.140,6.148]}$	**5.278** (0.021)	$5.245_{[5.241,5.249]}$
CDS	input	40	5.581[2](0.025)	$5.532_{[5.528,5.536]}$	5.396 (0.024)	$5.351_{[5.347,5.355]}$
CDS	input	80	5.416[2](0.029)	$5.367_{[5.363,5.371]}$	5.338 (0.032)	$5.281_{[5.277,5.285]}$
CDS	input	160	5.364[2](0.031)	$\mathbf{5.334}_{[5.330,5.339]}$	5.325 (0.034)	$5.279_{[5.275,5.283]}$
CDS	embd	40	5.518[1](0.007)	$5.508_{[5.504,5.512]}$	5.345 (0.006)	$5.336_{[5.332,5.340]}$
CDS	embd	80	5.405[1](0.047)	$5.370_{[5.365,5.374]}$	**5.278** (0.031)	$\mathbf{5.235}_{[5.231,5.239]}$
CDS	embd	160	**5.354**[1](0.004)	$5.350_{[5.346,5.354]}$	5.332 (0.005)	$5.325_{[5.321,5.330]}$

[1] N=3 (three seeds for optimization)

[2] N=9 (three seeds for optimization and three for symbolization)

[3] N is the same as for training.

A.6 Distributional Properties of Compositional Data in Symbolic Space

Table 9 shows the distributional properties of compositionally synthesized data by computing the pairwise Hellinger distance between symbolic representations of original training data (Tr), original test data (Te), and synthesized data (S). Given two discrete probability distributions $P = (p_1, \ldots, p_k)$ and $Q = (q_1, \ldots, q_k)$, the Hellinger distance is computed by the Euclidean norm of the difference of the square root vectors:

$$H(P, Q) = \frac{1}{\sqrt{2}} ||\sqrt{P} - \sqrt{Q}||_2. \tag{4}$$

Our experiments show that the distance between original train and original test distributions is an order of magnitude smaller for unigrams than for bigrams or trigrams, and the same relations hold for the distributional distance between synthesized data and original test data. These distributional properties — similar distributions between train and test data on the level of elementary components, here unigrams, and different distributions on the level of compounds, here higher order n-grams — are desired for benchmark data for compositional generalization (Keysers et al., 2020), and are recreated by our compositional data synthesization method.

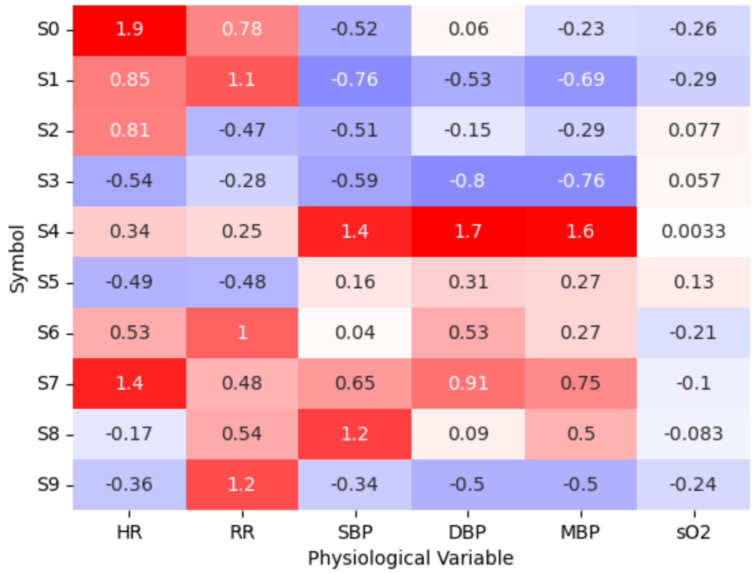
Table 9: Hellinger distance between distributions of symbolic representations of compositionally synthesized data based on MIMIC-III time series. Symbolization was performed by a random assignment of centroids in input space.

#Syms	**Unigram**			**Bigram**			**Trigram**		
	H(Tr, Te)	H(S, Te)	H(S, Tr)	H(Tr, Te)	H(S, Te)	H(S, Tr)	H(Tr, Te)	H(S, Te)	H(S, Tr)
40	0.0908	0.0895	0.0764	0.1612	0.1436	0.3438	0.4007	0.2482	0.2482
80	0.0237	0.0692	0.0597	0.1488	0.1886	0.1180	0.5231	0.5401	0.2845
160	0.0348	0.0616	0.0435	0.2712	0.2820	0.1185	0.7271	0.7142	0.3445

A.7 Qualitative Interpretation of Symbolic Time Series Representations

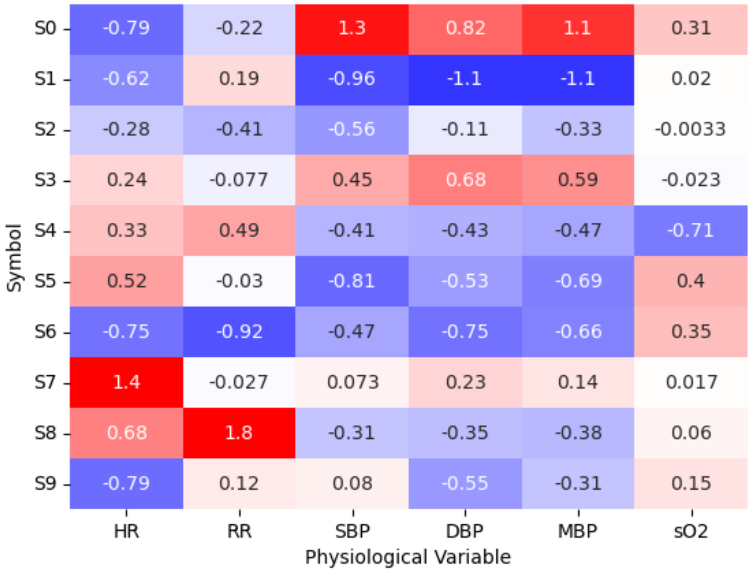
The goal of this qualitative evaluation is to investigate whether the learned clusters of subsequences of clinical time series can be interpreted as meaningful physiological states. We created dataset versions of MIMIC-III and eICU that are restricted to six features that are measured with high frequency (HR, SBP, DBP, MBP, RR and O2 Saturation). Based on these low-sparsity features, we performed *k*-means clustering on 3-hour blocks of time series subsequences, and assigned symbols to the clusters.

Figure 4 presents the results for *k*-means clustering of neural representations on MIMIC-III data, with *k* set to 10. Cluster S0 is characterized by elevated heart rate values and can be interpreted as representing the physiological state of tachycardia. Cluster S4 can he interpreted to represent the physiological state of hypertension due to elevated systolic, diastolic, and mean blood pressure. Similar results are obtained by computing clusters based on random centroids in the input domain of MIMIC-III. Figure 5 shows clusters representing tachycardia (S7), hypertension (S0), and tachypnea (S8) due to elevated respiratory rate.



	HR	RR	SBP	DBP	MBP	sO2
S0	1.9	0.78	-0.52	0.06	-0.23	-0.26
S1	0.85	1.1	-0.76	-0.53	-0.69	-0.29
S2	0.81	-0.47	-0.51	-0.15	-0.29	0.077
S3	-0.54	-0.28	-0.59	-0.8	-0.76	0.057
S4	0.34	0.25	1.4	1.7	1.6	0.0033
S5	-0.49	-0.48	0.16	0.31	0.27	0.13
S6	0.53	1	0.04	0.53	0.27	-0.21
S7	1.4	0.48	0.65	0.91	0.75	-0.1
S8	-0.17	0.54	1.2	0.09	0.5	-0.083
S9	-0.36	1.2	-0.34	-0.5	-0.5	-0.24

Figure 4: Physiological states learned by time series symbolization based on clustering of neural representations. For example, cluster S0 is characterized by elevated heart rate (HR), representing the physiological state of tachycardia. Cluster S4 can he interpreted to represent the physiological state of hypertension due to elevated systolic, diastolic, and mean blood pressure (SBP, DBP, MBP).



Symbol	HR	RR	SBP	DBP	MBP	sO2
S0	-0.79	-0.22	1.3	0.82	1.1	0.31
S1	-0.62	0.19	-0.96	-1.1	-1.1	0.02
S2	-0.28	-0.41	-0.56	-0.11	-0.33	-0.0033
S3	0.24	-0.077	0.45	0.68	0.59	-0.023
S4	0.33	0.49	-0.41	-0.43	-0.47	-0.71
S5	0.52	-0.03	-0.81	-0.53	-0.69	0.4
S6	-0.75	-0.92	-0.47	-0.75	-0.66	0.35
S7	1.4	-0.027	0.073	0.23	0.14	0.017
S8	0.68	1.8	-0.31	-0.35	-0.38	0.06
S9	-0.79	0.12	0.08	-0.55	-0.31	0.15

Figure 5: Physiological states learned by time series symbolization based on clustering in input space. Tachycardia is represented by cluster S7, showing elevated HR values. Hypertension is represented by cluster S0, showing elevated BP values. Cluster S8 represents the physiological state of tachypnea due to elevated respiratory rate (RR).