# SEQUENTIAL DISENTANGLED REPRESENTATION LEARNING BY DUAL-TRANSFORMER SUBTRACTION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Sequential disentanglement, which decomposes temporal observations into time-invariant (static) and time-varying (dynamic) factors, is fundamental to interpretable representation learning. We present DTS (Disentanglement by Dual-Transformer Subtraction), a simple framework that combines an architectural inductive bias for static–dynamic separation. DTS employs a hierarchical U-Net of dual-stream Transformers: at each level, a static stream without positional encoding runs in parallel with a position-encoded full stream, and their difference is a dynamic residual feeding the next level. We show that each level's static code is permutation-invariant by construction and that, under a weak-dependence assumption, its mean-pooled estimate concentrates at rate $O(1/\sqrt{T})$. With this architecture, a compact objective suffices: reconstruction with an optional second pass and at most one covariance regularizer. On six video, speech, and time-series benchmarks, DTS achieves the competitive results with the strongest baselines, fully unsupervised and modality-agnostic.

## 1 INTRODUCTION

Disentangled representation learning (DRL) for sequential observations is a core problem in modern representation learning (Li & Mandt, 2018). The goal is to factorize observations into time-invariant factors, such as identity, speaking style, or sensor-specific characteristics, and time-varying factors, such as motion, content, or temporal dynamics. Learning such representations can enhance interpretability (Hsu et al., 2017), support controllable generation and manipulation (Tian et al., 2021), and improve downstream tasks such as speaker verification and video prediction (Berman et al., 2023). These benefits are particularly pronounced in audio, video, and multivariate time series, where multiple latent factors interact and change over time (Barami et al., 2025).

Most existing approaches to sequential DRL follow the variational formulation of the Disentangled Sequential Variational Autoencoder (DSVAE) (Li & Mandt, 2018), and improve disentanglement by incorporating auxiliary modules or additional supervision signals (Bai et al., 2021; Han et al., 2021; Zhu et al., 2020; Luo et al., 2022; Tonekaboni et al., 2022; Naiman et al., 2023). Although effective in some settings, these methods often rely on increasingly elaborate model designs and heavily engineered objectives. In practice, they typically add extra loss terms, such as contrastive or adversarial objectives, or additional architectural constraints, each accompanied by additional hyperparameters that must be tuned to trade off reconstruction quality, temporal consistency, and factor separation. As a result, training becomes more complex and computationally expensive. For instance, S3VAE (Zhu et al., 2020) supplements the original VAE objective with three extra loss terms, each with its own balancing weight, and C-DSVAE (Bai et al., 2021) requires five hyperparameters solely to balance its loss terms (Zisling et al., 2026).

Recent efforts have reduced the complexity of sequential disentanglement but mostly remain within the VAE framework. Structured Koopman autoencoders (Berman et al., 2023) extend the problem from a static–dynamic decomposition to a more general multi-factor setting, although on speech their empirical results remain behind competitive DSVAE-based baselines (Naiman et al., 2023). Simon et al. (2024) model the static–dynamic dependence with a conditional normalizing flow, and DBSE (Berman et al., 2024) recovers the static posterior from a single observation and subtracts its features from the rest of the sequence. While effective, these methods still build on the DSVAE-based formulations, relying on additional assumptions, architectural components, or nontrivial



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

objective design. DiffSDA (Zisling et al., 2026) instead offers a diffusion-based alternative with less hyperparameter tuning and improved sample quality, but it inherits the substantial computational complexity of diffusion models. A simple and efficient framework for sequential disentanglement therefore remains an open need.

In this work, we present **DTS** (Disentanglement by **D**ual-**T**ransformer **S**ubtraction), a simple framework for sequential disentanglement that replaces variational inference with an architecture designed for static and dynamic separation. DTS is built on a hierarchical U-Net of dual-stream Transformers, in which a static stream without positional encoding operates in parallel with a full stream with positional encoding, and their difference defines the dynamic residual. This subtraction provides an inductive bias toward separating time-invariant and time-varying factors. Beyond the architectural design, we justify this inductive bias theoretically: each level's static code is permutation-invariant to its input over time by construction and its mean-pooled estimate converges at rate $O(1/\sqrt{T})$ under a weak-dependence assumption. Compared with prior methods, DTS requires only reconstruction with an optional dynamic-only second pass and at most one covariance regularizer, a much simpler objective. As the disentanglement operates in latent space, the framework is inherently modality-agnostic and combines directly with encoders and decoders for video, audio, and time-series data.

## 2 Related Work

Disentangled representation learning aims to separate observations into distinct factors of variation with minimal supervision. In the static setting, early work established both variational and adversarial formulations for disentanglement. VAE-based methods include $\beta$-VAE (Higgins et al., 2017), FactorVAE (Kim & Mnih, 2018), $\beta$-TCVAE (Chen et al., 2018), DIP-VAE (Kumar et al., 2018), and JointVAE (Dupont, 2018). In parallel, GAN-based approaches such as InfoGAN (Chen et al., 2016) promote interpretable latent factors by maximizing mutual information between latent codes and generated samples. Related work has also examined identifiability and the limits of unsupervised disentanglement (Hyvarinen & Morioka, 2016; Locatello et al., 2019; Khemakhem et al., 2020). While effective for static data, these methods do not account for temporal structure.

A standard extension to sequential data is DSVAE (Li & Mandt, 2018), which factorizes each sequence into a time-invariant latent variable and time-varying latent variables. Many subsequent methods build on this formulation by introducing additional objectives or structural constraints, including contrastive regularization in C-DSVAE (Bai et al., 2021), self-supervised and mutual-information objectives in S3VAE (Zhu et al., 2020), divergence penalties on the latents in R-WAE (Han et al., 2021), and posterior-sampled contrastive views in SPYL (Naiman et al., 2023). Related sequence models have also been developed for specific modalities, including content–motion disentanglement in video (Tulyakov et al., 2018; Denton & Birodkar, 2017; Villegas et al., 2017), speaker–content separation in speech (Hsu et al., 2017; Qian et al., 2019), and structured latent representations for time series (Fortuin et al., 2020; Tonekaboni et al., 2022). Although these methods often improve empirical disentanglement, they typically achieve it through increasingly elaborate objectives, which make optimization more difficult and more sensitive to dataset-specific tuning.

Recent work has sought to reduce this complexity. Berman et al. (2023) pursue multi-factor disentanglement by structured latent dynamics. DBSE (Berman et al., 2024) infers the static code from the first observation alone. Its dynamic code comes from an LSTM applied to the remaining observations after the first observation's encoder features have been subtracted from each of them. DiViD (Gheisari & Genovesio, 2025) adopts the similar idea for video diffusion. Simon et al. (2024) enrich the VAE formulation with a conditional normalizing flow. More recently, DiffSDA (Zisling et al., 2026) departs from the VAE family by formulating sequential disentanglement in a diffusion framework. In parallel, recent benchmark work has emphasized that real-world sequential disentanglement often extends beyond a simple static–dynamic split and requires evaluation in richer multi-factor settings (Barami et al., 2025), for which a simple and modality-agnostic building block is a natural starting point.

## 3 Methodology

Let $\mathbf{x}_{1:T} = \{x_1, \ldots, x_T\}$ denote a multivariate sequence of length $T$, where $x_t$ is a single observation. Sequential disentanglement seeks to decompose each sequence into a *static code* $s$ that captures time-invariant factors (e.g., speaker identity, character appearance) and a *dynamic code* $d_{1:T}$ that captures time-varying factors (e.g., phonemic content, actions, temporal dynamics), such that the two codes are statistically independent and can be recombined to reconstruct the input.

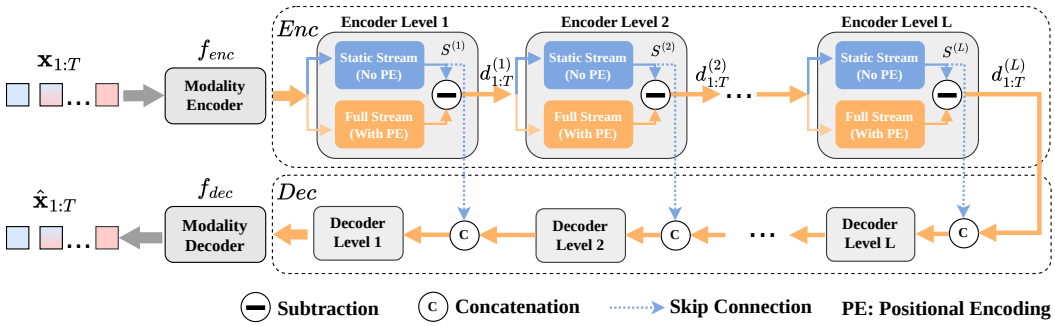


Figure 1: DTS architecture. Blue arrows are the static codes of each U-Net level, orange arrows the dynamic codes cascading through the encoder into the decoder, and dashed lines the skip connections.

DTS is built on a single architectural principle, *subtracting a permutation-invariant estimate from a position-aware stream*, and two types of loss terms: a reconstruction objective and a covariance regularization (CovReg) penalty. The subtraction mechanism is embedded in a hierarchical U-Net of dual-stream Transformers. It provides an inductive bias toward separating static and dynamic factors, and the static code is permutation-invariant by construction. CovReg enforces two properties of an ideally disentangled representation, i.e., agreement of the static code across temporal windows and zero cross-covariance between the static and dynamic codes, through a single penalty on second-order statistics. Since the disentanglement operates entirely in the latent space, it is modality-agnostic: only the modality-specific encoder $f_{\text{enc}}$ and decoder $f_{\text{dec}}$ change across domains.

### 3.1 Dual-Stream Subtraction Architecture

The core idea behind DTS is to isolate temporal dynamics by subtracting the time-invariant code through the architecture. An overview of DTS is shown in Figure 1. Specifically, a modality-specific encoder first maps each observation $x_t$ to a latent vector $g_t$. The resulting sequence $\mathbf{g}_{1:T}$ is then processed by a hierarchical U-Net of $L$ dual-stream encoder levels, each of which performs the following operations on its input $\mathbf{h}^{(l)}_{1:T}$ (with $\mathbf{h}^{(1)} = \mathbf{g}$).

**Static stream.** A Transformer encoder without positional encoding extracts the time-invariant code,

$$\tilde{s}^{(l)}_{1:T} = \mathrm{TF}^{(l)}_{\text{static}}\big(\mathbf{h}^{(l)}_{1:T}\big), \qquad s^{(l)} = \frac{1}{T}\sum_{t=1}^{T} \tilde{s}^{(l)}_t.$$

The absence of positional encoding is crucial: without position information, the above design yields a permutation-invariant output, that is reordering the observations in time leaves $s^{(l)}$ unchanged. We formalize this property in Section 3.5 (Proposition 1) and further show that the root-mean-square error of the static code decreases with $T$ at rate $O(1/\sqrt{T})$ (Lemma 1).

**Full stream.** A second Transformer encoder, equipped with learnable positional embeddings, processes the same input:

$$f^{(l)}_{1:T} = \mathrm{TF}^{(l)}_{\text{full}}\big(\mathbf{h}^{(l)}_{1:T} + \mathrm{PE}_{1:T}\big) \tag{1}$$

Because this stream has access to temporal position, it captures *both* static and dynamic information.

**Subtraction.** The dynamic residual is defined as the difference between the full representation and the static code:

$$d^{(l)}_t = f^{(l)}_t - s^{(l)} \tag{2}$$

This is the core inductive bias: the dynamic code is *architecturally defined* as the residual of the full stream after the static estimate is removed. When a static stream dimension $D_s$ differs from the full stream dimension $D$, we first map the static code through a learned linear projection $P^{(l)}_s : \mathbb{R}^{D_s} \to \mathbb{R}^D$. For readability, Eq. (2) absorbs this projection into the notation for $s^{(l)}$. This dimensional-alignment detail is important because it limits static capacity while preserving a well-defined subtraction operation.

**Hierarchical cascading.** The dynamic output of level $l$ becomes the input of level $l$+1: $\mathbf{h}^{(l+1)} = d^{(l)}_{1:T}$. This creates a progressive peeling structure: level 1 removes the coarsest static patterns, level 2 removes finer ones from the residual, and so on, producing multi-scale static codes $\{s^{(1)}, \ldots, s^{(L)}\}$ and a final dynamic code $d_{1:T} = d^{(L)}_{1:T}$.

**Decoder.** Reconstruction proceeds symmetrically from level $L$ back to 1. At each level, the reconstructed representation is combined with the corresponding static code via U-Net skip connections:

$$\hat{\mathbf{h}}_t^{(l-1)} = \mathrm{TF}_{\mathrm{dec}}^{(l)}\Big(\Phi\big([\hat{\mathbf{h}}_t^{(l)} \parallel s^{(l)}]\big)\Big) \tag{3}$$

where $[\cdot \parallel \cdot]$ is a concatenation operator and $\Phi$ is a linear projection. The recursion starts from $\hat{\mathbf{h}}^{(L)} = d_{1:T}$, and we denote the resulting U-Net decoder by $\mathrm{Dec}$.

### 3.2 Covariance Regularization

The subtraction architecture gives DTS a structural tendency to separate static and dynamic factors, but the model remains free to encode dynamic content redundantly into the static stream when that reduces the reconstruction error. To discourage such leakage, we propose a covariance regularization that directly penalizes deviations from the covariance structure of an ideally disentangled representation.

At the level of second-order statistics, an ideally disentangled representation has two properties. First, the static code is the same for every temporal window of a sequence, and its latent dimensions are mutually uncorrelated. In other words, the cross-covariance matrix between the standardized static codes of two temporal windows equals the identity $I$. Second, the static and dynamic codes have zero cross-covariance, which is a necessary consequence of statistical independence. CovReg penalizes deviations from both targets in a single expression:

$$\mathcal{L}_{\text{covreg}} = \frac{1}{L}\sum_{l=1}^{L}\Big[\big\|C_{ss}^{(l)} - I\big\|_{\mathrm{F}}^2 \;+\; \big\|C_{sd}^{(l)}\big\|_{\mathrm{F}}^2\Big]. \tag{4}$$

**Window agreement.** The static codes of two temporal windows at each level are obtained by splitting $\tilde{s}_{1:T}^{(l)}$ into temporal halves and mean-pooling each half separately:

$$z_a^{(l)} = \frac{1}{T_{1\mathrm{st}}}\sum_{t=1}^{T_{1\mathrm{st}}}\tilde{s}_t^{(l)}, \qquad z_b^{(l)} = \frac{1}{T - T_{1\mathrm{st}}}\sum_{t=T_{1\mathrm{st}}+1}^{T}\tilde{s}_t^{(l)},$$

where $T_{1\mathrm{st}} = \lfloor T/2 \rfloor$ is the length of the first half. The cross-covariance matrix $C_{ss}^{(l)} = \frac{1}{B}\,\bar{z}_a^{\top}\bar{z}_b$ is computed on $\bar{z}_a, \bar{z}_b$, the batch-standardized $z_a, z_b$ over $B$ sequences. Driving its diagonal toward 1 enforces agreement of the static code across the two windows, while driving off-diagonal entries toward 0 decorrelates latent dimensions and prevents representational collapse (Zbontar et al., 2021).

**Static dynamic independence.** $C_{sd}^{(l)}$ is the cross-covariance matrix between the static code $s^{(l)}$ and the final dynamic code $d_t$, computed over all pairs $(s^{(l)}, d_t)$ for every sequence in the mini-batch and every time step $t$. The penalty $\|C_{sd}\|_{\mathrm{F}}^2$ drives all pairwise covariances between the two codes toward zero. Together, both terms are squared deviations of an empirical matrix from its ideal target, making CovReg a single, coherent second-order penalty.

### 3.3 Reconstruction Loss

The primary training objective in DTS is *reconstruction*. Let $\mathcal{L}(\hat{\mathbf{x}}, \mathbf{x})$ denote a modality-specific reconstruction criterion. Given the decoder output $\hat{\mathbf{x}}_{1:T}$, the standard reconstruction term is

$$\mathcal{L}_{\text{recon}} = \mathcal{L}(\hat{\mathbf{x}}_{1:T},\, \mathbf{x}_{1:T}). \tag{5}$$

This term operates on the normal decoding path in which both static and dynamic codes are available through the U-Net skip connections.

**Dynamic-only reconstruction.** $\mathcal{L}_{\text{recon}}$ enforces the decoder to reproduce the input from both codes, but it does not say which code carries what content. Ideal separation requires two things: the dynamic code should not carry static content, and the static codes should not carry time-varying content. The first requirement is supported by the architecture, because the subtraction removes the static estimate from the dynamic path and CovReg decorrelates the two codes. The second requirement has no such support. The reason is that a static code can describe the dynamics without describing their order. The static stream sees every observation of the sequence, and the unordered set of observations often reveals *what* happened even if not *when*, e.g., the pose in a clip identifies the motion and the variance of a signal says how much it fluctuated. Such an order-free statistics of the dynamics is constant over the sequence and therefore passes the window-agreement term of CovReg unpenalized. More importantly, it reaches the decoder directly through the skip connections, where using it lowers $\mathcal{L}_{\text{recon}}$.

Since nothing in the objective penalizes such a summary while the reconstruction loss rewards it, the static codes may tend to absorb time-varying content.

We remove this tendency with a *second decoder pass* where all static codes are replaced by zero:

$$\mathcal{L}_{\text{dyn}} = \mathcal{L}\Big(f_{\text{dec}}\big(\text{Dec}(d_{1:T}, \{s^{(l)} = \mathbf{0}\}_{l=1}^{L})\big),\ \mathbf{x}_{1:T}\Big).$$

This pass must reconstruct the input from the dynamic code alone. The two passes gives:

$$\mathcal{L}_{\text{rec}} = \mathcal{L}_{\text{recon}} + \lambda_{\text{dyn}}\, \mathcal{L}_{\text{dyn}}. \tag{6}$$

$\mathcal{L}_{\text{dyn}}$ may seem to work against separation, since a dynamic code that reconstructs the full input must also carry static content. This concern misreads its purpose. Keeping static content out of the dynamic code is the job of the subtraction and CovReg. $\mathcal{L}_{\text{dyn}}$ addresses the opposite direction. It forces the dynamic code to carry all time-varying content by itself. As a result, the static codes are needed only for static content, for two reasons. First, once the dynamic code can explain the time-varying content by itself, copying that content into the static codes no longer reduces $\mathcal{L}_{\text{recon}}$, and nothing pushes it there. Second, the static codes have limited capacity, and they are a more accurate source of static content than the dynamic code (Lemma 1). The decoder therefore learns to take static content from the static codes whenever they are present, and time-varying content from the dynamic code.

### 3.4 Learning Objective

The complete DTS training objective combines Eq. (4) and Eq. (6).

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \lambda_{\text{dyn}}\, \mathcal{L}_{\text{dyn}} + \lambda_{\text{covreg}}\, \mathcal{L}_{\text{covreg}}. \tag{7}$$

At most two loss weights, $\lambda_{\text{covreg}}$ and $\lambda_{\text{dyn}}$, require tuning, in contrast to the two to five interdependent weights of prior methods (Zhu et al., 2020; Han et al., 2021; Bai et al., 2021). In summary, disentanglement in DTS rests on the dual-stream subtraction architecture, which keeps static content out of the dynamic code by subtracting the static estimate from the full stream at every level. The two loss terms complement this base where needed. CovReg refines both codes at second order. Its window-agreement term makes the static code identical across windows, and its cross-covariance term removes any remaining linear dependence between the static and dynamic codes. $\mathcal{L}_{\text{dyn}}$ keeps time-varying content out of the static codes by making the dynamic code sufficient to reconstruct the input on its own.

### 3.5 Architectural Properties

We next state what DTS does and does not ensure: a construction guarantee (Proposition 1), an estimation rate (Lemma 1), the effect of CovReg (Proposition 2), and the choice of the subtracted estimate (Proposition 3). Proofs and modeling assumptions are given in Appendix A.1, and scope and failure modes are discussed in Appendix A.4.

**Definition 1.** A representation $s^{(l)}$ computed from a sequence $\mathbf{h}_{1:T}$ is *temporally invariant* if for every permutation $\pi$ of $\{1, \ldots, T\}$,

$$s^{(l)}\big(\mathbf{h}_{\pi(1)}, \ldots, \mathbf{h}_{\pi(T)}\big) = s^{(l)}\big(\mathbf{h}_1, \ldots, \mathbf{h}_T\big).$$

The above definition on temporal invariance ensures that permuting the observations in time leaves the representation unchanged, but it does not prevent time-varying content from being encoded through order-free statistics, such as the mean and variance of the observations.

**Proposition 1** (Permutation-invariant static code)**.** *Let $\text{TF}_{\textit{static}}$ be a deterministic Transformer encoder with shared self-attention and position-wise feed-forward layers, no positional encoding, and no position-dependent masks or token-specific parameters. If $s^{(l)} = \frac{1}{T}\sum_{t=1}^{T}[\text{TF}_{\textit{static}}(\mathbf{h}_{1:T})]_t$, then $s^{(l)}$ is temporally invariant in the sense of Definition 1.*

Proposition 1 establishes an architectural property of the static stream, which therefore holds for whatever weights training produces. The property is per level. For $l \geq 2$ it holds with respect to the level's own input, and end-to-end permutation invariance of deeper static codes is therefore an inductive bias.

**Lemma 1** (Static code estimation under weak dependence)**.** *Fix the static factor of a sequence and let $s^*$ be the conditional mean of $\tilde{s}_t^{(l)}$ given that static factor. Then $\tilde{s}_t^{(l)} = s^* + \epsilon_t$ with $\mathbb{E}[\epsilon_t] = 0$ over*

*sequences sharing that factor. Suppose* $\mathbb{E}[\|\epsilon_t\|_2^2] \le D\sigma^2$*, and for every* $t$ *and lag* $k \ge 1$ *there exist nonnegative coefficients* $\eta_k$*, independent of* $t$*, such that*

$$\left|\mathbb{E}[\epsilon_t^\top \epsilon_{t+k}]\right| \le D\sigma^2\eta_k.$$

*Then, with* $R_T = \sum_{k=1}^{T-1}(1 - k/T)\eta_k$*, the static code* $s^{(l)}$ *satisfies*

$$\mathbb{E}\big[\|s^{(l)} - s^*\|_2^2\big] \le \frac{D\sigma^2}{T}\big(1 + 2R_T\big). \tag{8}$$

*If* $\sum_{k\ge1}\eta_k < \infty$ *(e.g.,* $m$*-dependent or mixing perturbations), the mean-squared error is* $O(1/T)$ *and the root-mean-square error is* $O(1/\sqrt{T})$*.*

The lemma justifies the static stream as a stable estimator of what is constant over the sequence. It states that the output of the static stream, $s^{(l)}$, approaches $s^*$ as the sequence grows longer, provided that the deviations $\epsilon_t$ are weakly correlated across time. Two remarks follow. First, $R_T$ is set by the correlation time[1] of the deviations. The faster the correlations decay, the more accurate the static code. Appendix A.1 measures this correlation time on the trained models, and Figure A.1 confirms the predicted rate. Second, the lemma does not say what constant content the static code captures. It converges to whatever is constant over the sequence, including order-free summaries of the dynamics. $\mathcal{L}_{\text{dyn}}$ and CovReg then decide what constant content to keep.

**Proposition 2** (What CovReg enforces)**.** *For standardized* $z_a, z_b$*, static and dynamic codes* $s, d$*, the population CovReg objective is minimized when* $\mathrm{Cov}(z_a, z_b) = I$ *and* $\mathrm{Cov}(s, d) = 0$*. At the level where* $d_t = f_t - P_s s$*, substituting this definition gives* $\mathrm{Cov}(s, d_t) = \mathrm{Cov}(s, f_t) - \mathrm{Var}(s)P_s^\top$*, which vanishes exactly when* $P_s = \mathrm{Cov}(f, s)\,\mathrm{Var}(s)^{-1}$*, the least-squares coefficient for predicting* $f_t$ *from* $s$*. At the optimum, the subtracted term* $P_s s$ *is therefore the best linear prediction of the full stream from the static code, and the dynamic code* $d_t$ *is the regression residual, the part of* $f_t$ *that cannot be linearly predicted from* $s$*. Zero cross-covariance is equivalent to independence for jointly Gaussian codes and is a second-order surrogate otherwise.*

In practice, the first condition requires the static code computed from the first half of a sequence to agree with the one computed from the second half. In other words, the static code should not change along the sequence. The second condition determines what the subtraction removes. Because $P_s$ is learned jointly with the model, minimizing CovReg drives $P_s$ toward the regression coefficient of the proposition, and the full stream then decomposes as $f_t = P_s s + d_t$ with $d_t$ uncorrelated with $s$. This is the additive form that Proposition 3 assumes.

**Proposition 3** (Which static estimate to subtract)**.** *Assume the full stream has the aligned form* $f_t = P_s s^* + u_t + \xi_t$ *that Proposition 2 enforces at second order, where* $u_t$ *is the target dynamic component and* $\xi_t$ *is representation error, and let* $\bar{u} = \frac{1}{T}\sum_t u_t$ *be the window mean of the dynamics. Consider three estimates of the static component to subtract from* $f_t$*.*

1. The separate stream's estimate $P_s s$*, as in DTS. The residual* $\hat{d}_t = f_t - P_s s$ *satisfies*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big[\|\hat{d}_t - u_t\|_2^2\big] \le \frac{2}{T}\sum_{t=1}^{T}\mathbb{E}\big[\|\xi_t\|_2^2\big] + 2\|P_s\|_{\text{op}}^2\,\mathbb{E}\big[\|s - s^*\|_2^2\big], \tag{9}$$

   *which is* $O(\bar{\xi} + 1/T)$ *under Lemma 1.*

2. The temporal mean of the full stream $\bar{f}$*, as in the STS control of Section 4.3. The residual* $f_t - \bar{f}$ *has average error exactly* $\|\bar{u}\|_2^2$ *plus the centered variance of* $\xi$*.*

3. A single element $f_1$*, as in DBSE (Berman et al., 2024). The residual* $f_t - f_1$ *has average error* $\|u_1\|_2^2$ *plus terms in* $\xi$*.*

*Only DTS's error vanishes with* $T$*, while the other two have* $T$*-independent floors* $\|\bar{u}\|_2^2$ *and* $\|u_1\|_2^2$*.*

The proposition justifies the separate static stream. An estimate taken from the full stream itself, whether its temporal mean or one of its elements, carries part of the dynamics: the mean removes the window mean of the dynamics, and a single element removes its own dynamic state. Only an estimate from a separate, order-blind stream is free of both, and its error shrinks as sequences grow longer (Lemma 1), up to the representation error $\xi_t$ that reconstruction and CovReg reduce. Table 9 compares these three estimates empirically.

[1]The correlation time is the number of consecutive steps over which the deviation stays roughly the same before it changes, that is, how many steps one must move before $\epsilon_{t+k}$ has forgotten $\epsilon_t$.

# 4 EXPERIMENTS

## 4.1 EXPERIMENTAL CONFIGURATIONS

We evaluate DTS on six benchmarks spanning three modalities: video (Sprites (Reed et al., 2015)), audio (TIMIT (Garofolo et al., 1993) and LibriSpeech (Panayotov et al., 2015)), and time series (ETTh1 (Zhou et al., 2021), Air Quality (Zhang et al., 2017), PhysioNet (Goldberger et al., 2000)). We compare against multiple baselines: MoCoGAN (Tulyakov et al., 2018), FHVAE (Hsu et al., 2017), DSVAE (Li & Mandt, 2018), R-WAE (Han et al., 2021), S3VAE (Zhu et al., 2020), C-DSVAE (Bai et al., 2021), SKD (Berman et al., 2023), SPYL (Naiman et al., 2023), GLR (Tonekaboni et al., 2022), DBSE (Berman et al., 2024), and DiffSDA (Zisling et al., 2026). For fair comparison, we follow the evaluation protocol of the strongest prior work on each benchmark (per-benchmark details in Appendix A.2) and, whenever possible, quote numbers reported in the original papers.

DTS uses the same dual-stream subtraction architecture (Section 3) with modality-specific encoders and decoders across all datasets. The U-Net depth is $L = 3$ on all benchmarks and ablations on $L$ are in Section 4.3. The reconstruction criterion $\mathcal{L}$ is MSE for Sprites and ETTh1, L1 for TIMIT and LibriSpeech, and masked MSE for Air Quality and PhysioNet. The complete objective contains at most two tunable weights ($\lambda_{\text{covreg}}$, $\lambda_{\text{dyn}}$). Dataset-specific hyperparameters are listed in Appendix A.3.

## 4.2 EXPERIMENTAL RESULTS

We first compare DTS against all reported baselines on each benchmark dataset.

**Video disentanglement (Sprites).** Table 1 reports the standard action-swap evaluation (Zhu et al., 2020; Bai et al., 2021). As can be seen, DTS achieves 100% action accuracy and IS = 9.000, the maximum attainable under the nine-class action classifier used in this evaluation, with zero conditional entropy $H(y|x) = 0.000$. DTS reaches the theoretical IS bound, surpassing DBSE and SPYL (IS = 8.942) and slightly improving over SKD (IS = 8.999). It does so with only two loss terms and without task-label supervision during representation learning.

Table 1: Sprites action swap: action accuracy, IS, and prediction entropies of the judge.

| **Method** | **Acc↑** | **IS↑** | $\mathbf{H}(\mathbf{y}\|\mathbf{x})\downarrow$ | $\mathbf{H}(\mathbf{y})\uparrow$ |
|---|---|---|---|---|
| MoCoGAN | 92.89 | 8.461 | 0.090 | 2.192 |
| DSVAE | 90.73 | 8.384 | 0.072 | 2.192 |
| R-WAE | 98.98 | 8.516 | 0.055 | 2.197 |
| S3VAE | 99.49 | 8.637 | 0.041 | 2.197 |
| SKD | 100.0 | 8.999 | 0.000 | 2.197 |
| C-DSVAE | 99.99 | 8.871 | 0.014 | 2.197 |
| SPYL | 100.0 | 8.942 | 0.006 | 2.197 |
| DBSE | 100.0 | 8.942 | 0.006 | 2.197 |
| **DTS (ours)** | **100.0** | **9.000** | **0.000** | **2.197** |

Table 1 scores only the factor that is *kept* (the action) and never shows whether the injected static code actually determines the appearance. Table 2 closes this gap with paired swaps in both directions. The *action swap* decodes a sequence's dynamic code with the static code of a randomly paired sequence while the *content swap* does the reverse. For each of the five factors (the 9-class action and four 6-class appearance attributes), a pre-trained classifier scores the generated clip against the *donor* that supplied the code carrying that factor (transfer, higher is better) and against the *replaced* sequence whose code was discarded (leakage, lower is better). Leakage exists only for the factor whose code was swapped out, hence the dashes, and chance is $1/9{=}11.1\%$ for action and $1/6{=}16.7\%$ per attribute. Three observations follow. First, appearance follows the injected static code: skin, top, and hair transfer at 99.6–100% and pant at 93.9–94.0%, in both directions. Second, appearance leakage from the replaced sequence is 17.7–18.6%, within two points of chance, so a swapped-out static code contributes essentially nothing to the output. Third, action follows the dynamic code at 97.4–97.5%, with a residual 14.5% match to the replaced sequence (chance 11.1%): a little action information remains in the static path, but not enough to override the injected dynamic code. Table 2 therefore shows what Table 1 cannot: DTS routes action into the dynamic code and appearance into the static code simultaneously, with negligible leakage through the other path.

Table 2: Sprites paired swaps: each factor scored against the sequence supplying its code (transfer) and against the replaced sequence (leakage).

| | **Action swap** | | **Content swap** | | |
|---|---|---|---|---|---|
| **Factor** | transf.↑ | leak↓ | transf.↑ | leak↓ | **Chance** |
| Action | 97.5 | – | 97.4 | 14.5 | 11.1 |
| Skin | 99.6 | 17.8 | 99.6 | – | 16.7 |
| Pant | 93.9 | 18.6 | 94.0 | – | 16.7 |
| Top | 100.0 | 17.7 | 100.0 | – | 16.7 |
| Hair | 100.0 | 18.6 | 100.0 | – | 16.7 |

**Audio disentanglement.** Tables 3 and 4 compare speaker verification performance (Hsu et al., 2017) on TIMIT and LibriSpeech, where disentanglement is measured by the gap between dynamic

Table 5: ETTh1 MAE and PhysioNet mortality.

| Method | ETTh1↓ | AUPRC↑ | AUROC↑ |
|---|---|---|---|
| C-DSVAE | $12.53 \pm 0.88$ | $0.16 \pm 0.01$ | $0.57 \pm 0.01$ |
| GLR | $12.27 \pm 0.03$ | $0.37 \pm 0.09$ | $0.75 \pm 0.01$ |
| SPYL | $12.22 \pm 0.03$ | $0.37 \pm 0.02$ | $0.76 \pm 0.04$ |
| DBSE | $11.21 \pm 0.01$ | $0.47 \pm 0.02$ | $0.86 \pm 0.01$ |
| DiffSDA | $9.89 \pm 0.28$ | $\mathbf{0.50 \pm 0.006}$ | $\mathbf{0.87 \pm 0.004}$ |
| **DTS (ours)** | $\mathbf{8.267 \pm 0.45}$ | $\mathbf{0.50 \pm 0.011}$ | $0.83 \pm 0.004$ |

Table 6: Air Quality and ICU-type accuracy.

| Method | Air Quality↑ | ICU type↑ |
|---|---|---|
| C-DSVAE | $47.07\% \pm 1.20$ | $32.54\% \pm 0.00$ |
| GLR | $50.32\% \pm 3.87$ | $38.93\% \pm 2.48$ |
| SPYL | $57.93\% \pm 3.53$ | $46.98\% \pm 3.04$ |
| DBSE | $65.87\% \pm 0.01$ | $56.87\% \pm 0.34$ |
| DiffSDA | $69.2\% \pm 1.50$ | $\mathbf{64.6\% \pm 0.35}$ |
| **DTS (ours)** | $\mathbf{81.25\% \pm 2.29}$ | $62.55\% \pm 0.44$ |

and static equal error rate (EER). On TIMIT, DTS reaches a gap of 48.02%, exceeding DBSE (31.11%) by +16.9 pp and DiffSDA (42.29%) by +5.7 pp, with the lowest static EER (1.80%) and a dynamic EER near chance under this protocol (49.82%). The gain comes from both sides of the gap. The static EER is roughly half that of the best baseline (SPYL, 3.41%), which indicates that speaker identity is concentrated more fully in the static code. The dynamic EER is the highest among all methods (49.82% vs. 46.72% for DiffSDA), which indicates that the dynamic code is left to carry the phonetic content that varies within an utterance. On LibriSpeech DTS reaches 37.56% (static 9.05%, dynamic 46.62%), exceeding DiffSDA (34.93%) by +2.6 pp. The gain over DiffSDA mainly comes from the static side. DTS's static EER is 2.0 pp lower (9.05% vs. 11.02%), while its dynamic EER matches DiffSDA's within 0.7 pp (46.62% vs. 45.94%). Both dynamic codes therefore carry a similar small residual of speaker information. Static EERs are higher on LibriSpeech than on TIMIT for every method, which reflects the larger speaker pool and more varied recordings of the audiobook corpus, and DTS keeps the lowest static EER under these harder conditions. The VAE baselines stay far behind, with gaps of 5.86–24.89%. SPYL's chance-level dynamic EER comes at the cost of a 24.87% static EER.

Table 3: TIMIT speaker verification (EER).

| Method | Stat.↓ | Dyn.↑ | Gap↑ |
|---|---|---|---|
| FHVAE | 5.06% | 22.77% | 17.71% |
| DSVAE | 5.64% | 19.20% | 13.56% |
| R-WAE | 4.73% | 23.41% | 18.68% |
| S3VAE | 5.02% | 25.51% | 20.49% |
| SKD | 4.46% | 26.78% | 22.32% |
| C-DSVAE | 4.03% | 31.81% | 27.78% |
| SPYL | 3.41% | 33.22% | 29.81% |
| DBSE | 3.50% | 34.62% | 31.11% |
| DiffSDA | 4.43% | 46.72% | 42.29% |
| **DTS (ours)** | **1.80%** | **49.82%** | **48.02%** |

Table 4: LibriSpeech speaker verification (EER).

| Method | Stat.↓ | Dyn.↑ | Gap↑ |
|---|---|---|---|
| DSVAE | 15.06% | 28.94% | 13.87% |
| DBSE | 16.75% | 22.61% | 5.86% |
| SPYL | 24.87% | **49.76%** | 24.89% |
| DiffSDA | 11.02% | 45.94% | 34.93% |
| **DTS (ours)** | **9.05%** | 46.62% | **37.56%** |

**Time series.** Tables 5 and 6 report the results on the time series benchmarks, following DBSE and DiffSDA. The tasks are oil-temperature regression (MAE) on ETTh1, month classification on Air Quality from the static codes, and mortality prediction (AUPRC/AUROC) and ICU-type classification on PhysioNet. DTS sets the best result on ETTh1 (MAE 8.267, 16.4% below DiffSDA and 26.3% below DBSE) and on Air Quality (81.25%, +12.05 pp over DiffSDA and +15.38 pp over DBSE). These wins support the central claim of DTS. Both targets are sequence-level structure of exactly the kind the static stream is built to hold, namely the slowly varying thermal state of the transformer and the seasonal and station baselines behind the month label. The dual-stream hierarchy separates them from day-level fluctuations without any task-specific loss. The Air Quality baseline ordering points the same way. Methods with an explicit static-inference constraint (SPYL, DBSE, DiffSDA) lead C-DSVAE and GLR by 8–22 pp, and DTS extends the trend with a constraint built into the architecture rather than the objective PhysioNet shows where the static–dynamic trade-off lies. DTS matches DiffSDA on AUPRC (0.50) and reaches 62.55% ICU accuracy, above every VAE-based baseline (DBSE 56.87%) and within 2.1 pp of DiffSDA, while trailing on AUROC (0.83 vs. 0.87). Mortality depends on acute pre-death trajectories that favor richer dynamic modeling, whereas ICU type reflects ward-level factors that a single subtraction level preserves well. The depth analysis in Section 4.3 shows that a shallower model improves both tasks further.

### 4.3 Property Analysis

This section analyzes the U-Net depth, the loss components, and the subtraction architecture of DTS on PhysioNet, ETTh1, and TIMIT (Tables 7–9), varying one property at a time.

**U-Net depth.** We show the results in Table 7 where we vary the U-Net depth $L \in \{1, \ldots, 5\}$. DTS is relatively robust to depth. Every TIMIT depth exceeds DiffSDA's gap by at least 3.8 pp with the dynamic EER near chance. On ETTh1, $L$=1 to $L$=4 all beat both baselines. On PhysioNet, every depth stays at or above DBSE, and $L$=1 leads DiffSDA on both tasks. The results indicate that the preferred depth may depend on the amount and complexity of the static content in the input. Each level subtracts one more static code. Simpler static content, such as speaker identity and ward-level clinical state, is captured by one level, whereas the richer static content of the long ETTh1 windows profits from three. Depth is thus an interpretable knob. The shared $L$=3 is competitive everywhere, and using a suitable depth adds further gains. The ablations below use the best depth from Table 7 for each benchmark, which measures the contribution of each component against the strongest model.

Table 7: U-Net depth $L$ (*default).

| $L$ | PhysioNet | | ETTh1 | TIMIT (%) | | |
|---|---|---|---|---|---|---|
| | AUPRC↑ | ICU↑ | MAE↓ | Stat.↓ | Dyn.↑ | Gap↑ |
| 1 | **0.51** | **65.09** | 9.130 | **1.41** | **49.87** | **48.46** |
| 2 | 0.50 | 57.69 | 9.858 | 1.80 | 49.12 | 47.32 |
| 3 | 0.50* | 62.55* | **8.267*** | 1.80* | 49.82* | 48.02* |
| 4 | 0.48 | 63.25 | 8.723 | 2.17 | 48.50 | 46.33 |
| 5 | 0.47 | 62.35 | 12.199 | 3.60 | 49.70 | 46.10 |
| DBSE | 0.47 | 56.87 | 11.21 | 3.50 | 34.62 | 31.11 |
| DiffSDA | 0.50 | 64.6 | 9.89 | 4.43 | 46.72 | 42.29 |

**Loss components.** Table 8 shows the ablation results of removing $\mathcal{L}_{\text{covreg}}$, $\mathcal{L}_{\text{dyn}}$, or both from the full objective. The results show that removing CovReg degrades all three benchmarks sharply. $\mathcal{L}_{\text{dyn}}$ brings a further gain on both PhysioNet tasks, and removing both terms degrades PhysioNet more than removing either alone. The results further indicate that reconstruction alone already achieves a good degree of disentanglement, which shows the effectiveness of the proposed dual-stream subtraction architecture. The reconstruction-only variant remains competitive with DBSE on all three benchmarks and exceeds it on the TIMIT gap. Overall, each loss term is useful on its own, and combining both with the dual-stream subtraction architecture further boosts disentanglement.

Table 8: Loss ablation (–: disabled.)

| Objective | PhysioNet | | ETTh1 | TIMIT (%) | | |
|---|---|---|---|---|---|---|
| | AUPRC↑ | ICU↑ | MAE↓ | Stat.↓ | Dyn.↑ | Gap↑ |
| Full | **0.51** | **65.09** | **8.267** | **1.41** | **49.87** | **48.46** |
| $-\,\mathcal{L}_{\text{covreg}}$ | 0.49 | 52.79 | 12.52 | 13.18 | 46.70 | 33.52 |
| $-\,\mathcal{L}_{\text{dyn}}$ | 0.48 | 63.50 | – | – | – | – |
| $\mathcal{L}_{\text{recon}}$ only | 0.44 | 50.75 | 12.52 | 13.18 | 46.70 | 33.52 |

**Dual-stream subtraction architecture.** Table 9 ablates the two ideas of the dual-stream subtraction architecture with matched-capacity controls. *DT* keeps both streams but drops the subtraction, taking the full-stream output directly as the dynamic code, and *STS* keeps the subtraction but replaces the learned static estimate by the temporal mean of the full stream. The results show that DTS is the best variant on all six metrics. Without the subtraction, DT degrades on every metric, as nothing then keeps the two codes complementary. STS is also below DTS on all six metrics. Subtracting the temporal mean removes not only the static content but also the average of the dynamics over the window, which is the error floor identified in Proposition 3. This hurts most on ETTh1, whose target is itself a window-level average, whereas the learned static stream removes only what is stable across the sequence. DBSE, which subtracts a single element, trails DTS on every metric, in line with the error floors of Proposition 3. These results indicate that both a permutation-invariant static estimate (Proposition 1) and its subtraction are required, and identical losses cannot compensate for either.

Table 9: Architecture ablation.

| Method | PhysioNet | | ETTh1 | TIMIT (%) | | |
|---|---|---|---|---|---|---|
| | AUPRC↑ | ICU↑ | MAE↓ | Stat.↓ | Dyn.↑ | Gap↑ |
| DTS | **0.51** | **65.09** | **8.267** | **1.41** | **49.87** | **48.46** |
| DT | 0.41 | 56.87 | 13.65 | 4.38 | 38.10 | 33.72 |
| STS | 0.43 | 60.07 | 13.01 | 3.58 | 48.02 | 44.44 |
| DBSE | 0.47 | 56.87 | 11.21 | 3.50 | 34.62 | 31.11 |

## 5 Conclusion

We presented DTS, a modality-agnostic framework that disentangles sequences through an architectural inductive bias, a permutation-invariant static stream whose estimate is subtracted from a full stream, paired with a compact two-term objective. Theoretical analyses justify this design by showing that the static stream is a good estimate of the static code and the right quantity to subtract from the full stream. Empirically, DTS achieves state-of-the-art results on six video, speech, and time-series benchmarks, and the ablations trace these gains to each component of the model. These results show that a well-chosen inductive bias can replace much of the objective engineering in sequential disentanglement, and establish DTS as a promising method for this task.

## AI USE STATEMENT

Generative AI tools were used in preparing this manuscript. A large language model assistant (Claude, Anthropic) was used to edit the text for readability, to restructure LaTeX tables and page layout, to cross-check the numbers in the main text against the archived result files, to check the correctness of the mathematical derivations, to verify and format the bibliographic references, and to give feedback on the clarity and organization of the paper. All AI-assisted text and checks were reviewed and revised by the authors, and every reported number was verified against the original experiment logs and checkpoints. Generative AI tools were not used to generate or alter experimental data or results. The authors take full responsibility for the content of this paper.

## REFERENCES

Junwen Bai, Weiran Wang, and Carla P Gomes. Contrastively disentangled sequential variational autoencoder. In *Advances in Neural Information Processing Systems*, volume 34, pp. 10105–10118, 2021.

Tal Barami, Nimrod Berman, Ilan Naiman, Amos H Hason, Rotem Ezra, and Omri Azencot. Disentanglement beyond static vs. dynamic: A benchmark and evaluation framework for Multi-Factor sequential representations. In *Advances in Neural Information Processing Systems*, volume 38, pp. 140328–140370, 2025.

Nimrod Berman, Ilan Naiman, and Omri Azencot. Multifactor sequential disentanglement via structured Koopman autoencoders. In *International Conference on Learning Representations*, 2023.

Nimrod Berman, Ilan Naiman, Idan Arbiv, Gal Fadlon, and Omri Azencot. Sequential disentanglement by extracting static information from a single sequence element. In *International Conference on Machine Learning*, pp. 3539–3564. PMLR, 2024.

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Emily L Denton and Vighnesh Birodkar. Unsupervised learning of disentangled representations from video. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Emilien Dupont. Learning disentangled joint continuous and discrete representations. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. GP-VAE: Deep probabilistic time series imputation. In *International Conference on Artificial Intelligence and Statistics*, pp. 1651–1661. PMLR, 2020.

John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, David S Pallett, Nancy L Dahlgren, and Victor Zue. TIMIT acoustic-phonetic continuous speech corpus LDC93S1. Linguistic Data Consortium, 1993.

Marzieh Gheisari and Auguste Genovesio. DiViD: Disentangled video diffusion for static-dynamic factorization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 341–350. IEEE, 2025.

Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

Jun Han, Martin Renqiang Min, Ligong Han, Li Erran Li, and Xuan Zhang. Disentangled recurrent Wasserstein autoencoder. In *International Conference on Learning Representations*, 2021.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. $\beta$-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

Wei-Ning Hsu, Yu Zhang, and James Glass. Unsupervised learning of disentangled and interpretable representations from sequential data. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Aapo Hyvarinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ICA: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pp. 2207–2217. PMLR, 2020.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pp. 2649–2658. PMLR, 2018.

Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2018.

Yingzhen Li and Stephan Mandt. Disentangled sequential autoencoder. In *International Conference on Machine Learning*, pp. 5670–5679. PMLR, 2018.

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 *International Conference on Machine Learning*, pp. 4114–4124. PMLR, 2019.

Yin-Jyun Luo, Sebastian Ewert, and Simon Dixon. Towards robust unsupervised disentanglement of sequential data—a case study using music audio. In *International Joint Conference on Artificial Intelligence*, pp. 3299–3305, 2022.

Ilan Naiman, Nimrod Berman, and Omri Azencot. Sample and predict your latent: Modality-free sequential disentanglement via contrastive estimation. In *International Conference on Machine Learning*, pp. 25694–25717. PMLR, 2023.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210. IEEE, 2015.

Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson. AutoVC: Zero-shot voice style transfer with only autoencoder loss. In *International Conference on Machine Learning*, pp. 5210–5219. PMLR, 2019.

Scott E Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Mathieu Cyrille Simon, Pascal Frossard, and Christophe De Vleeschouwer. Sequential representation learning via static-dynamic conditional disentanglement. In *European Conference on Computer Vision*, pp. 110–126. Springer, 2024.

Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. In *International Conference on Learning Representations*, 2021.

Sana Tonekaboni, Chun-Liang Li, Sercan O Arik, Anna Goldenberg, and Tomas Pfister. Decoupling local and global representations of time series. In *International Conference on Artificial Intelligence and Statistics*, pp. 8700–8714. PMLR, 2022.

Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing motion and content for video generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1526–1535, 2018.

Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. Learning to generate long-term future via hierarchical prediction. In *International Conference on Machine Learning*, pp. 3560–3569. PMLR, 2017.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow Twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021.

Shuyi Zhang, Bin Guo, Anlan Dong, Jing He, Ziping Xu, and Song Xi Chen. Cautionary tales on air-quality improvement in Beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2205):20170457, 2017.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient Transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115, 2021.

Yizhe Zhu, Martin Renqiang Min, Asim Kadav, and Hans Peter Graf. S3VAE: Self-supervised sequential VAE for representation disentanglement and data generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6537–6546, 2020.

Hedi Zisling, Ilan Naiman, Nimrod Berman, Supasorn Suwajanakorn, and Omri Azencot. DiffSDA: Unsupervised diffusion sequential disentanglement across modalities. In *International Conference on Learning Representations*, 2026.