Plasma State Monitoring and Disruption Characterization using Multimodal VAEs

Yoeri Poels[1,2], Alessandro Pau[1], Christian Donner[3], Giulio Romanelli[3], Olivier Sauter[1], Cristina Venturini[1], Vlado Menkovski[2], the TCV team[4] and the WPTE team[5]

[1]École Polytechnique Fédérale de Lausanne (EPFL), Swiss Plasma Center (SPC), CH-1015 Lausanne, Switzerland
[2]Eindhoven University of Technology (TU/e), Mathematics and Computer Science, NL-5600MB Eindhoven, The Netherlands
[3]Swiss Data Science Center (SDSC), ETH Zürich & EPFL, CH-8092 Zürich & CH-1015 Lausanne, Switzerland
[4]See author list of B. P. Duval *et al.* 2024 *Nucl. Fusion* **64** 112023
[5]See author list of E. Joffrin *et al.* 2024 *Nucl. Fusion* **64** 112019

E-mail: yoeri.poels@epfl.ch

April 2025

Abstract

When a plasma disrupts in a tokamak, significant heat and electromagnetic loads are deposited onto the surrounding device components. These forces scale with plasma current and magnetic field strength, making disruptions one of the key challenges for future devices. Unfortunately, disruptions are not fully understood, with many different underlying causes that are difficult to anticipate. Data-driven models have shown success in predicting them, but they only provide limited interpretability. On the other hand, large-scale statistical analyses have been a great asset to understanding disruptive patterns. In this paper, we leverage data-driven methods to find an interpretable representation of the plasma state for disruption characterization. Specifically, we use a latent variable model to represent diagnostic measurements as a low-dimensional, latent representation. We build upon the Variational Autoencoder (VAE) framework, and extend it for (1) continuous projections of plasma trajectories; (2) a multimodal structure to separate operating regimes; and (3) separation with respect to disruptive regimes. Subsequently, we can identify continuous indicators for the disruption rate and the disruptivity based on statistical properties of measurement data. The proposed method is demonstrated using a dataset of approximately 1600 TCV discharges, selecting for flat-top disruptions or regular terminations. We evaluate the method with respect to (1) the identified disruption risk and its correlation with other plasma properties; (2) the ability to distinguish different types of disruptions; and (3) downstream analyses. For the latter, we conduct a demonstrative study on identifying parameters connected to disruptions using counterfactual-like analysis. Overall, the method can adequately identify distinct operating regimes characterized by varying proximity to disruptions in an interpretable manner.

1. Introduction

A disruption in a tokamak corresponds to the rapid loss of plasma control. The plasma is no longer confined, terminating the discharge and rapidly depositing the plasma's thermal and electromagnetic energy onto the surrounding vessel. This phenomenon can cause severe machine damages [1, 2], especially in future devices [3, 4], making disruptions one of the key challenges for the exploitation of tokamaks in future power plants [5]. Unfortunately, the exact physical processes involved in disruptions are not fully understood, many different trajectories can lead to disruptive conditions, and it is not fully determined how to best predict their onset [6–10].

Past experiments have generated vast datasets capturing both disruptive and non-disruptive plasma behavior. Large-scale statistical analyses have been a cornerstone for the understanding of disruptions [9–11]. Simultaneously, expressive statistical methods, such as deep neural networks (NNs) [12], have shown much success in the task of predicting disruptions [13–19]. However, these disruption prediction methods often only provide limited interpretability. Instead, we aim to leverage similar expressive statistical methods with an explicit focus on finding interpretable representations of the plasma operational space, rather than directly mapping observables to a prediction in a black-box manner.

We propose a method for plasma state monitoring using a *latent variable model*. Here, the goal is to find an abstract, low-dimensional representation that models a large quantity of discharges, each defined by a time series of signal data. In other words, we aim to automatically identify an abstract phase space of a tokamak's operational space. Specifically, we leverage machine learning methods to learn the transformation from data space to this so-called latent space, which is simultaneously optimized to accurately represent the original measurements while providing separability w.r.t. disruptive regimes. We propose the method as a complement to prediction methods, instead focusing on enhancing analysis and understanding.

We leverage the framework of Variational Autoencoders (VAEs) [20, 21]. A VAE models a data distribution $p(x)$ under the assumption that it is generated by a latent variable z, and models the relationship between data space and latent space using conditional distributions parametrized by neural networks. We adapt this framework with three

properties in mind: (1) a continuous projection of sequential measurements—for tracking discharges as they evolve; (2) a multimodal latent variable—for separating the operational space into discrete regimes; all while (3) providing separability w.r.t. disruptive regimes. We implement (1) using a dynamic formulation for the encoder distribution [22], modeling the time derivative of the latent variable. For (2), we utilize a multimodal, Gaussian mixture-based prior distribution for z [23, 24]. To improve training dynamics, we propose extensions to the model architecture and its optimization. Finally, (3) is achieved by learning a 'disruption risk' variable as a function of z.

Due to the smooth structure of the learned manifold and its separability w.r.t. disruptive dynamics, we can use it to identify continuous analogues of the disruption rate and the disruptivity [7, 9] (as detailed in Section 2). Additionally, in this initial study, we constrain z to be 2-dimensional, allowing for easier interpretability of the manifold and its relation to physical parameters. The proposed method is demonstrated with a dataset of approximately 1600 TCV discharges. We characterize flat-top dynamics, consequently selecting for flat-top disruptions or regular terminations, and additionally filter on discharges reaching a diverted X-point configuration. TCV provides a challenging testbed given its large variability in plasma scenarios [25] and its sensitivity to fast disruptions connected to MHD limits [26–28].

We validate the identified disruption risk variable both quantitatively and qualitatively. Specifically, we evaluate its calibration w.r.t. the disruption rate, compare it to the disruptivity, and identify its correspondence to known operational limits. In this context, we evaluate the ability to automatically distinguish clusters of distinct types of disruptions, as found in ITER Baseline Scenario (IBL) experiments [29], density limits [30] and negative triangularity scenarios [31]. Additionally, we correlate the identified states in latent variable z with plasma properties such as the confinement state [32] and disruption precursors [26]. Lastly, we conduct a proof-of-principle study of using the latent variable model to automatically identify disruption-related parameters, by identifying counterfactual pairs of (non-)disrupting discharges and the times-of-interest therein.

Adjacent latent variable-based methods have utilized Generative Topographic Mapping [33–35] and Self-Organizing Maps (SOM) [36–38] to characterize disruptive regions in the tokamak operational space. In contrast, we utilize the VAE for its expressivity in learning the latent variable utilizing neural networks, alongside its flexibility in shaping the optimization target. Closest to our setting are [39] and [40], both also building upon the VAE framework. We differentiate through an explicit focus on learning a multimodal, state-based representation, allowing for better clustering of distinct operating regimes. Notably, there are also ongoing efforts towards automated detection of chains-of-events connected to disruptions [26, 41]. We consider the proposed method complementary to such efforts, providing an extra tool to this end. In short, our contributions can be summarized as follows:

- We develop a method to automatically identify a low-dimensional, multimodal, interpretable representation of the operational space of TCV, optimized to identify operational limits w.r.t. disruptions.
  - We build upon existing VAE-based approaches, and propose extensions to the model architecture and its optimization to encourage multimodality in the learned posterior distribution.
  - The method automatically clusters distinct regimes, and can be used to identify continuous analogues for the disruption rate and disruptivity of these parameter spaces.
- We build a database of approximately 1600 TCV discharges containing either (a) flat-top disruptions or (b) regular terminations, covering a representative sample of the TCV operational space.
- We extensively evaluate the proposed method both quantitatively and qualitatively. We evaluate (1) latent space properties and their connection to disruption metrics and events; (2) the clustering of distinct operating regimes; and (3) using the model for downstream analyses. For the latter, we conduct a proof-of-principle study of identifying disruption-related parameters using counterfactual-like analysis.

2. Problem Formulation

Disruptions in tokamaks are known to be caused by a wide array of phenomena [9, 10, 42]. Various operational limits have been identified, such as the Greenwald limit for the plasma density [43] or the low-q current limit (Kruskal-Shafranov limit), relating to the ratio of toroidal to poloidal magnetic field components [44–46]. Many different plasma conditions are associated with heightened risk of instabilities eventually leading to a disruption [6, 7]. Statistical analyses of past disruptions can aid the understanding of disruptive boundaries and disruption causes [10], and simultaneously present the opportunity for learning a plasma state representation for integration into advanced control schemes [47]. Consequently, the automatic identification of patterns present in large datasets of disrupting discharges can aid efforts towards better understanding of disruptions and contribute to achieving more robust device operations. We aim to learn a nonlinear low-dimensional representation of high-dimensional observations, efficiently compressing down the vast amount of experimental data to its core patterns.

We define the problem setting as modeling a dataset of tokamak discharges $\mathbf{x} \in X \subseteq \mathbb{R}^{U \times T^{in}}$, for U input signals and T^{in} time samples. These discharges are modeled using a latent variable $\mathbf{z} \in \mathbb{R}^{d \times T^z}$ of d dimensions, for latent

trajectories of T^z timesteps. A prior distribution is assumed for $p(\mathbf{z})$, allowing us to model the data distribution $p(\mathbf{x})$ as $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$. Since we model the data as timesteps, we aim to map realizations of latent variable $\mathbf{z}$ to signals at a single timestep t_m, i.e., we want to model:

$$p(\mathbf{x}^{t_m}|\mathbf{z}^{t_m}), \tag{1}$$

for a *decoder* distribution p mapping from latent space to data space[1]. However, we do not want to treat individual timesteps as independent, but rather model the plasma state as a trajectory in the latent space. Consequently, we aim to learn a distribution where each latent state depends on the previous state (Markov assumption):

$$q(\mathbf{z}^{t_1}, \ldots, \mathbf{z}^{t_m}|\mathbf{z}^{t_0}, \mathbf{x}^{\mathbf{t}\leq t_m}) = \prod_{i=1}^{m} q(\mathbf{z}^{t_i}|\mathbf{z}^{t_{i-1}}, \mathbf{x}^{\leq t_i}), \tag{2}$$

for an *encoder* distribution q mapping from data space to latent space for the signals up to and including timestep t_m. Combined, we model the full distribution $p(\mathbf{x}, \mathbf{z})$ using sequences of latent trajectories to represent sequences of signals in data space.

In addition to representing a signal time series as a trajectory of a latent state, we aim to shape latent variable $\mathbf{z}$ with additional desiderata. For interpretability reasons we set $d = 2$, i.e. a 2-dimensional latent space, to allow for visualization of the entire latent space at once; higher dimensionalities will be studied in future works. To allow for differentiating distinct regimes in the latent state, we require $\mathbf{z}$ to be structured for clustering:

$$\mathcal{C} : \mathbf{z} \in \mathbb{R}^d \rightarrow \{1, \ldots, K\}, \tag{3}$$

where $\mathcal{C}$ denotes a mapping of the latent variable to one of K latent clusters.

Finally, variable $\mathbf{z}$ should be informative w.r.t. the plasma's proximity to disruptions. We introduce a variable indicating this notion as an average risk of disruption:

$$D_{\text{risk}} : \mathbf{z} \in \mathbb{R}^d \rightarrow [0, 1]. \tag{4}$$

To quantify D_{risk}, we aim for it to represent the fraction of shots that eventually disrupt[2]. We can consequently interpret it as a continuous analogue of the disruption rate, where rather than selecting groups of shots a priori, we exploit the smooth manifold of learned latent variable $\mathbf{z}$ to automatically define the disruption rate for different parameter spaces.

Note that we do not model explicitly how close in time we are to a disruption. Since we aim to model the entire operational space rather than the chain-of-events inevitably leading to a disruption, and given that we do not take into account future control actions, predicting a future disruption is ill-posed. Rather, D_{risk} is used to organize the latent space into different regions associated with disruptions to find common patterns and to provide an average level of risk. It is also related, but not equivalent, to the notion of disruptivity, defined as the number of disruptions per second for a given plasma property space [7, 10]. We can project a continuous equivalent of the disruptivity onto the latent space, which we denote as $\hat{D}_{\text{disr}}$. We additionally compare D_{risk} to $\hat{D}_{\text{disr}}$.

[1]We omit explicit indexing on the signals for $\mathbf{x}$, i.e. $\mathbf{x}^{\mathbf{u}}$, for conciseness, since we always operate on all input signals.

[2]Since this quantity is defined using the organization of plasma regimes by a learned, unknown latent variable, we do not learn the rate directly in a supervised setting. However, post-hoc analysis can be done on learned variable D_{risk} to calibrate it and to quantify whether it can accurately represent the disruption rate.

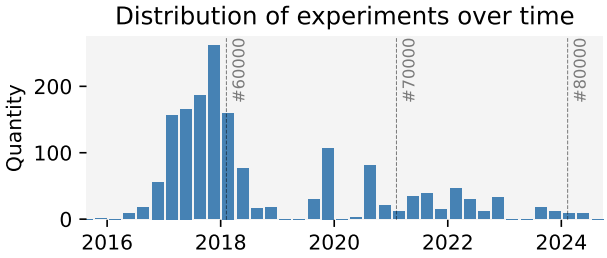

Figure 1: The distribution of the discharges' dates, binned per quartile. Discharges range from TCV #51325 to #81751, with the majority taking place between 2016-2019.

3. Dataset

We utilize a dataset of approximately 1600 TCV discharges, ranging from TCV #51325 (Dec 2015) to #81751 (Jun 2024), see Figure 1 for the distribution of shots over time. To illustrate the variety of plasma scenarios we plot the distribution of key parameters in Figure 2.

3.1. Dataset construction

The aim of this work is to characterize plasma dynamics in the flat-top phase of the discharge. The ramp-up and ramp-down phases are not considered here due to their non-stationary nature and distinct plasma trajectories, and are left for future investigation. To cover a representative sample of the TCV flat-top operational space, we construct the dataset by sampling thousands of past experiments and then filter on discharges that either (a) disrupt in the flat-top phase or (b) have a regular termination. Additionally, we constrain the scope to plasmas that reach a lower single null (X-point) configuration at some point during the flat-top phase. The dataset statistics are summarized as follows:

- 1629 discharges ranging from #51325 to #81751
- 1768.96 s of flat-top plasma dynamics
  - On average, $1.086\,\text{s} \pm 0.49\,\text{s}$ per shot
  - 297.53 s with limited, 1471.43 s with diverted configuration
- 1147 flat-top disruptions, 482 regular terminations
  - Disruption rate of $70.4\,\%$[3]

Variable	Unit	Description
A_p	m^2	Plasma cross-sectional area
Δ_{min}	m	Minimum radial gap between the plasma edge and the inner or outer wall
δ_{bottom}		Lower (bottom) plasma triangularity
δ_{top}		Upper (top) plasma triangularity
κ		Plasma elongation
$\text{AXUV}_{\text{X-point}}$	a.u.	Emissions measured using AXUV diodes covering the X-point region
$\text{PD}^{\textit{CIII}}_{\text{FFT}}$	a.u.	Spectral distribution of photodiode signal for CIII line emission (λ=465.1 nm), computed as the variance of frequency spectra for windows of $20\,\mathrm{ms}$
I_p	A	Plasma current
l_i		Internal inductance of the plasma current
q_{95}		Safety factor at 95% of enclosed magnetic flux
$n_{e,\text{core}}$	m^{-2}	Vertical interferometer line-integrated electron density from 0.87 m < ch < 0.91 m
n_e/n_{GW}		Greenwald fraction [43] using electron density measurements from interferometry
SXR_{core}	$\mathrm{W\,m}^{-1}$	Soft X-Ray core ($\rho_\psi < 0.15$) emission
P_{in}	W	Total input power
W_{tot}	J	Total plasma stored energy
β_N		Normalized toroidal beta ($100 \cdot \beta_t \frac{aB_0}{I_p\ [\text{MA}]}$)
LM	T	Locked mode amplitude
$RMS_{(n=1)}$	T	Root-mean-square of the $n = 1$ mode amplitude for windows of $2\,\mathrm{ms}$
$RMS_{(n=2)}$	T	Root-mean-square of the $n = 2$ mode amplitude for windows of $2\,\mathrm{ms}$
γ_{VGR}	Hz	Vertical growth rate estimate using RZIp [27]

Table 1: The list of input signals and constructed features used in the latent variable model. Equilibrium-related features originate from LIUQE [48], MHD markers are computed using fast magnetic probes [49].

The dataset consists of a time series for each shot, where features correspond to diagnostic measurements or features derived from them, see Subsection 3.2 for more details. Additionally, it contains disruption-related metadata, such as the time of disruption t_D. We define t_D as the onset of the thermal quench leading to the eventual disruption, and automatically compute these times using the DEFUSE framework [26]. As extra metadata used for the evaluation of the identified latent space, we leverage event detectors from the DEFUSE framework to populate the database[4]. For example, events related to MHD (magnetohydrodynamics) markers for rotating/locked modes or indicators related to abnormal kinetic profiles (e.g. peaking or hollowing thereof). Finally, we leverage automated tools for confinement state classification [32] for evaluation, to correlate the identified states with different confinement modes. Specifically, we use the full ensemble presented in [32] to predict all confinement states and use detections with at least 75% prediction confidence. These predictions cover $1533.84\,\mathrm{s}$ of plasma dynamics: $1287.18\,\mathrm{s}$ L-mode (83.9%), $16.56\,\mathrm{s}$ of dithering (1.1%) and $230.10\,\mathrm{s}$ H-mode (15.0%).

[3]Note that the dataset is biased towards disruptive plasmas, and 70.4% does not necessarily reflect the disruption rate of TCV operation.

[4]The development and evaluation of the automated event detection in DEFUSE is still ongoing. Nevertheless, early evaluations indicate that the aggregate distributions of detections are statistically meaningful.

3.2. Signals

The utilized signals are selected either because they are representative of plasma scenarios and/or because of their connection to known operational limits. An overview of all signals is provided in Table 1. The signals are interpolated to a common timebase of $10\,\mathrm{kHz}$ using the last available sample—we refrain from non-causal interpolation methods (e.g., linear, spline) to ensure no information leakage about future plasma dynamics. Data retrieval is handled through the DEFUSE framework [26].

4. Method

The approach consists of a latent variable model that represents time series of discharge measurements as trajectories in a latent space. This latent space should be structured to distinguish different operating regimes, while being connected to disruption characteristics. We extend the Variational Autoencoder (VAE) framework [20, 21], taking into account aforementioned desiderata. We introduce the VAE framework in Section 4.1, our structure in Section 4.2, the training and inference procedure in Section 4.3, followed by details on the architecture in Section 4.4.

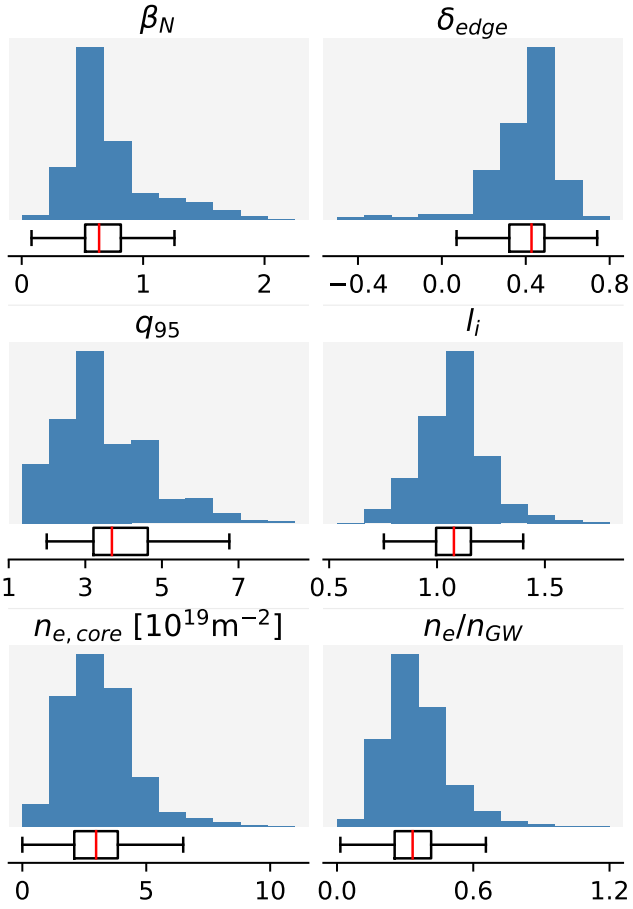

Figure 2: Distributions of key plasma parameters in the dataset. The plotted values are averages over phases of 20 ms–around the TCV energy confinement time–to exclude transient states.

4.1. The variational autoencoder

The Variational Autoencoder is a generative model that aims to model a data distribution $p(\mathbf{x})$ under the assumption that it is generated using latent variable $\mathbf{z}$. We consequently model $p(\mathbf{x})$ as $p(\mathbf{x},\mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$, choose prior distribution $p(\mathbf{z})$ and learn the data distribution as $p(\mathbf{x}|\mathbf{z})$. In the VAE framework, we additionally learn approximate posterior distribution $q(\mathbf{z}|\mathbf{x})$. Distributions $q(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{x}|\mathbf{z})$ are often referred to as the encoder and decoder, respectively, given that they encode/decode from data space to latent space and vice versa.

The encoder and decoder distribution in a VAE are parametrized by neural networks (NNs). In the standard formulation, the encoder is parametrized as follows:

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}^2_\phi(\mathbf{x})I), \tag{5}$$

$$\boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}_\phi(\mathbf{x}) = f_\phi(\mathbf{x}), \tag{6}$$

for model parameters ϕ. For a Gaussian decoder, we can similarly parametrize the mean and variance. However, in practice, we often only parametrize the mean and fix the variance term:

$$p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\theta(\mathbf{z}), \sigma^2_\theta I), \tag{7}$$

$$\boldsymbol{\mu}_\theta(\mathbf{z}) = f_\theta(\mathbf{z}), \tag{8}$$

for model parameters θ. Both the encoder and decoder are jointly optimized using a lower bound on the log likelihood of the data, the evidence lower bound (ELBO):

$$\text{ELBO} := \ln p(\mathbf{x}) \geq \underbrace{\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}|\mathbf{z})]}_{-\mathcal{L}_{rec}(\mathbf{x})} \tag{9}$$

$$- \underbrace{\mathbb{E}[\ln q_\phi(\mathbf{z}|\mathbf{x}) - \ln p(\mathbf{z})]}_{\mathcal{L}_{KL}(\mathbf{z})}. \tag{10}$$

The ELBO term can be split into two parts. The former, $\mathcal{L}_{rec}$, can be considered as a reconstruction error. Usually, the expectation terms are approximated by sampling, which corresponds to taking data points and encoding and decoding them. Then, the error between the input samples and the reconstructions is minimized. The latter, $\mathcal{L}_{KL}$, can be considered a regularization term. It is equivalent to the Kullback–Leibler divergence [50], and expresses how well the learned latent distribution is covered by the prior distribution. Often, it is approximated by taking samples of $\mathbf{z}$ and computing the relative log-likelihoods.

In order to get stable gradient estimates while training, latent variable $\mathbf{z}$ in Equation 5 is sampled using the reparameterization trick [20]. We parametrize parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ of the distribution and sample using external noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ as $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$ (where $\odot$ denotes element-wise multiplication).

4.2. Sequential multimodal VAE with disruption risk

To better address the problem statement, we extend the VAE formulation to incorporate a notion of time, to model the disruption risk, and introduce a structure for clustering. We first introduce the extended formulation for all components, including the neural network functions we learn in practice. Then, we describe the prior distribution and its role in clustering in the latent space.

Components. Similar to recurrent VAEs [22, 51], we introduce the notion of time into the VAE by modeling a dependence between the latent variable at a given timestep, $\mathbf{z}^{t_m}$, and its past states, $\mathbf{z}^{\mathbf{t}<t_m}$, see also Equation 2. However, for efficiency reasons, we do not learn a dependence on all signals at each timestep. Rather, we operate using a fixed timewindow of size w, which we slide across input signals with stride s. Then, our encoding distribution is reformulated as follows:

$$\mathcal{T}_i = \{0, s, 2s, \ldots, m\}, \tag{11}$$

$$q(\mathbf{z}^{t_1}, \ldots, \mathbf{z}^{t_m}|\mathbf{z}^{t_0}, \mathbf{x}^{\mathbf{t}\leq t_m}) = \prod_{i\in\mathcal{T}_i} q(\mathbf{z}^{t_i}|\mathbf{z}^{t_{i-s}}, \mathbf{x}^{t_{i-w}:t_i}), \tag{12}$$

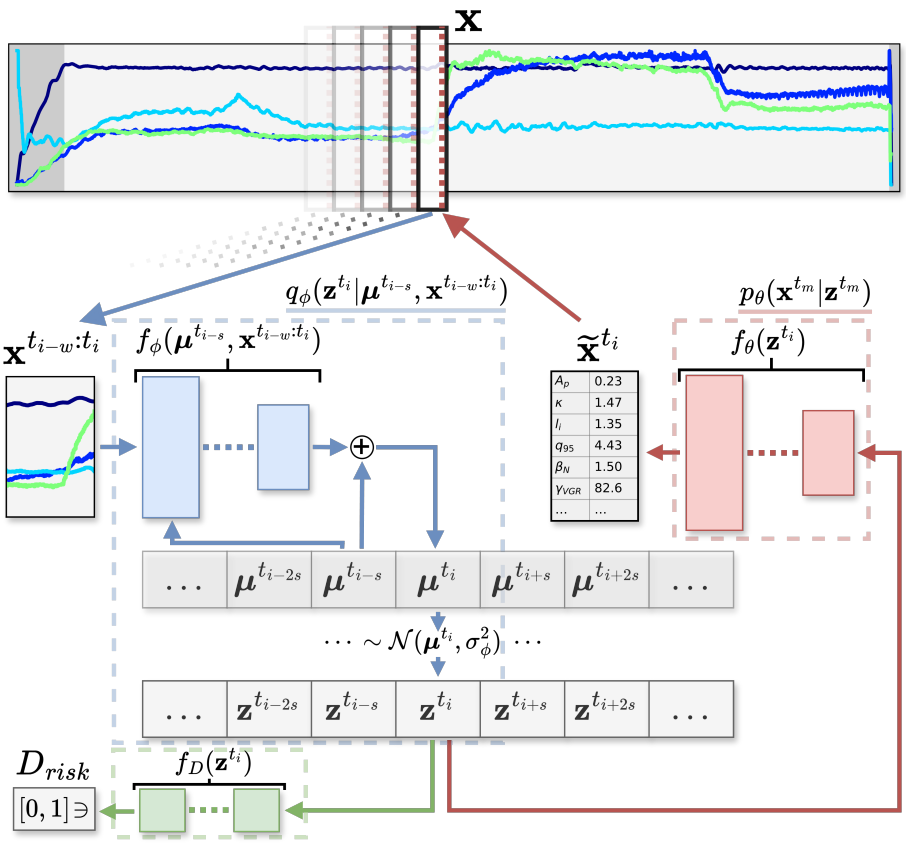

Figure 3: Schematic overview of the model structure. The model consists of encoder distribution $q_\phi(\mathbf{z}^{t_i}|\boldsymbol{\mu}^{t_{i-s}}, \mathbf{x}^{t_{i-w}:t_i})$ (blue), decoder distribution $p_\theta(\mathbf{x}^{t_i}|\mathbf{z}^{t_i})$ (red) and disruption risk map $D_{risk}(\mathbf{z}^{t_i})$ (green). Signals are encoded using timewindows of input signals, with the location in the latent space being computed as an update w.r.t. the previous location. Conversely, the mapping from latent space to data space and to the disruption risk are static in time.

assuming m as a multiple of s. That is, we model sequences at a sampling rate of $\frac{1}{s}$ using as input timewindows of signals of size w and the previous latent state. For the decoder, we model a single timestep, i.e.,

$$p_\theta(\mathbf{x}^{t_m}|\mathbf{z}^{t_m}), \tag{13}$$

for the plasma discharge at time t_m. The fixed mapping allows for interpretation of the latent variable by projecting it back directly to physics quantities in data space.

Both the encoder and decoder are parametrized by neural networks as follows. To encourage continuity in the latent space, we formulate the encoder distribution to update the location with respect to the previous timestep. To reduce noise while training, we utilize the distribution mean directly rather than sampling at each timestep in the trajectory, i.e., we approximate q as follows:

$$q(\mathbf{z}^{t_m}|\mathbf{z}^{t_{m-s}}, \mathbf{x}^{t_{m-w}:t_m}) \approx q_\phi(\mathbf{z}^{t_m}|\boldsymbol{\mu}^{t_{m-s}}, \mathbf{x}^{t_{m-w}:t_m}), \tag{14}$$

for parameters ϕ, where $\boldsymbol{\mu}^{t_{i-s}}$ is the mean of q_ϕ at the previous timestep. Then, the structure becomes:

$$\begin{aligned} q_\phi(\mathbf{z}^{t_m}|\boldsymbol{\mu}^{t_{m-s}}, \mathbf{x}^{t_{m-w}:t_m}) = \\ \mathcal{N}(\mathbf{z}^{t_m}; \boldsymbol{\mu}^{t_{m-s}} + \mu_\phi(\boldsymbol{\mu}^{t_{m-s}}, \mathbf{x}^{t_{m-w}:t_m}), \sigma_\phi^2 I), \end{aligned} \tag{15}$$

$$\mu_\phi(\boldsymbol{\mu}^{t_{m-s}}, \mathbf{x}^{t_{m-w}:t_m}) = f_\phi(\boldsymbol{\mu}^{t_{m-s}}, \mathbf{x}^{t_{m-w}:t_m}), \tag{16}$$

for neural network function f_ϕ. For stability during training [52] and fast projections of large sets of data distributions, we fix σ_ϕ as a model hyperparameter. Note that we parametrize the mean of q_ϕ using the previous mean summed to the neural network output. That is, we can consider the network as a coarse neural differential equation [53] that models the mean of the latent distribution, which is subsequently solved using a forward Euler scheme with fixed timestep s.

For the decoder distribution we parameterize the mean and fix the variance:

$$p_\theta(\mathbf{x}^{t_m}|\mathbf{z}^{t_m}) = \mathcal{N}(\mathbf{x}^{t_m}; \boldsymbol{\mu}_\theta(\mathbf{z}^{t_m}), \sigma_\theta^2 I), \tag{17}$$

$$\boldsymbol{\mu}_\theta(\mathbf{z}^{t_m}) = f_\theta(\mathbf{z}^{t_m}), \tag{18}$$

for neural network function f_θ. Additionally, we learn a

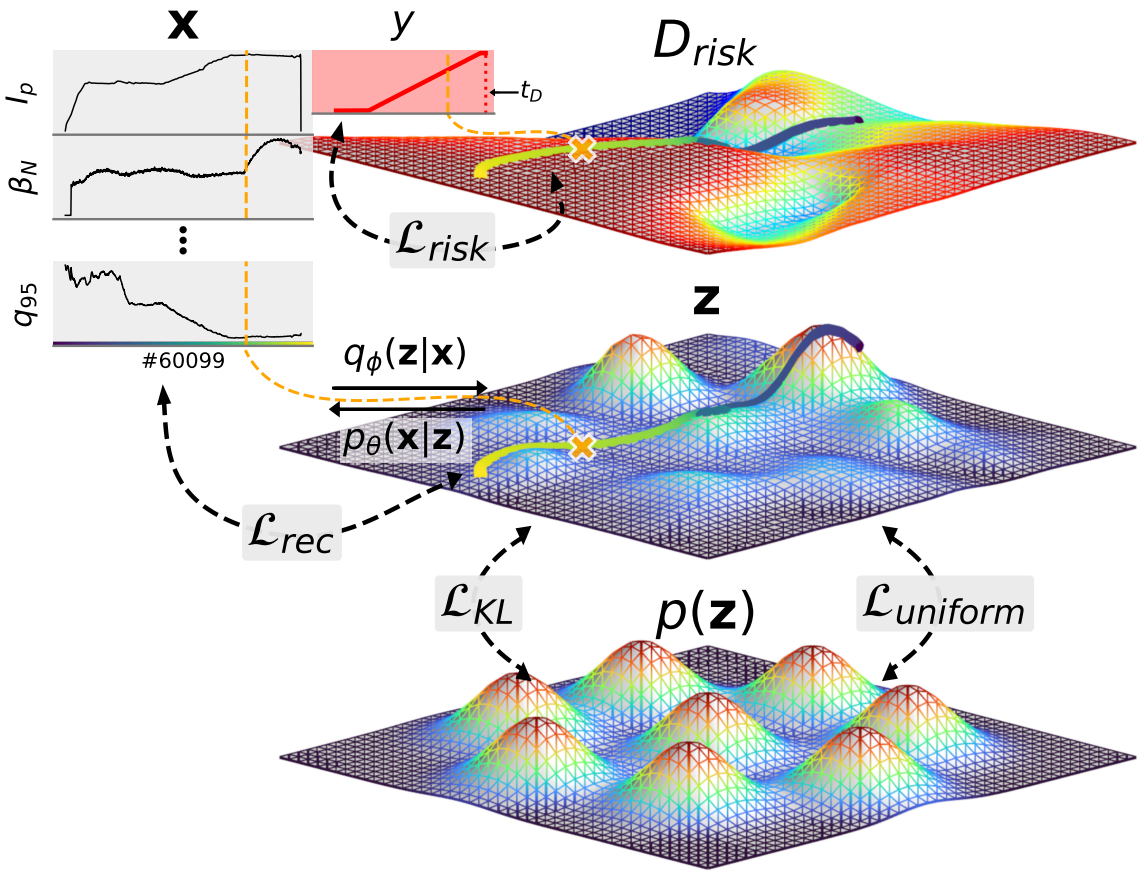

Figure 4: Depiction of the training and inference procedure of the proposed method. Data $\mathbf{x}$, time series of signals corresponding to TCV experiments, are projected to latent variable $\mathbf{z}$ using approximate posterior, the encoder, $q_\phi(\mathbf{z}^{t_m}|\boldsymbol{\mu}^{t_{m-s}}, \mathbf{x}^{t_{m-w}:t_m})$ (or $q_\phi(\mathbf{z}|\mathbf{x})$ aggregating over time). The generative distribution, the decoder, $p_\theta(\mathbf{x}^{t_m}|\mathbf{z}^{t_m})$ (or $p_\theta(\mathbf{x}|\mathbf{z})$ aggregating over time) provides the map back to data space. A timeslice in data space corresponds to a timeslice in latent space, consequently projecting discharges as latent trajectories. Simultaneously, we learn disruption risk map D_{risk} as a function of $\mathbf{z}$, using proxy labels y. The latent space is optimized to maximize the data likelihood ($\mathcal{L}_{rec}$), match disruption information ($\mathcal{L}_{risk}$), minimize divergence to prior modes ($\mathcal{L}_{KL}$) while covering all modes of said prior ($\mathcal{L}_{uniform}$).

disruption variable D_{risk} as a function of $\mathbf{z}$:

$$D_{risk}(\mathbf{z}^{t_m}) = f_D(\mathbf{z}^{t_m}), \tag{19}$$

for neural network function f_D. An overview of the model components and their interaction is given in Figure 3.

Multimodality. To introduce a structure suited for identifying different regimes, we formulate prior distribution $p(\mathbf{z})$ as a multimodal distribution. Specifically, we use a mixture of Gaussians:

$$p(\mathbf{z}) = \sum_{k=1}^{K} w_k \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_k, \sigma_k^2), \tag{20}$$

for a mixture of K Gaussian distributions parameterized by means $\boldsymbol{\mu}_k$, standard deviations $\boldsymbol{\sigma}_k$, and mixing weights w_k. Assuming sufficient distance between the distributions, $p(\mathbf{z})$ will have multimodal structure, encouraging our learned latent variable to similarly have multiple peaks.

For simplicity, we fix the standard deviation for all distributions, and choose equal mixing weights $w_k = \frac{1}{K}$. Then, we can define the likelihood of being assigned to a cluster as the likelihood under the Gaussian mixture:

$$\mathcal{C} := p(\mathcal{C} = k|\mathbf{z}) = \frac{\mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_k, \sigma_p^2)}{\sum_{j=1}^{K} \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_j, \sigma_p^2)}, \tag{21}$$

with fixed prior standard deviation σ_p.

4.3. Training and inference

To train NN functions f_ϕ (encoder), f_θ (decoder) and f_D (disruption risk), we extend the ELBO. Given an input

timewindow of signals $\mathbf{x}^{t_{m-w}:t_m}$, we sample the latent variable:

$$\mathbf{z}^{t_m} \sim q_\phi(\mathbf{z}^{t_m}|\boldsymbol{\mu}^{t_{m-s}}, \mathbf{x}^{t_{m-w}:t_m}), \tag{22}$$

whereas for the reconstruction, we directly use the mean of the decoder:

$$\widetilde{\mathbf{x}}^{t_m} = \boldsymbol{\mu}_\theta(\mathbf{z}^{t_m}). \tag{23}$$

Then, the reconstruction reduces to a squared error as follows:

$$\mathcal{L}_{rec}(\mathbf{x}^{t_m}) = (\widetilde{\mathbf{x}}^{t_m} - \mathbf{x}^{t_m})^2. \tag{24}$$

The regularization term is as follows:

$$\mathcal{L}_{KL}(\mathbf{z}) = KL(q_\phi(\mathbf{z}^{t_m}|\mathbf{z}^{t_{m-s}}, \mathbf{x}^{t_{m-w}:t_m}) \parallel p(\mathbf{z})), \tag{25}$$

which we estimate by using samples of $\mathbf{z}^{t_m}$ and computing the relative log-likelihood under both the prior and posterior. Since the KL divergence can act mode seeking in this setting [54, 55], we utilize it to encourage single samples of $\mathbf{z}$ to lie on one mode of the prior $p(\mathbf{z})$, and dynamically set its mixing weights w_k as $p(C = k|\mathbf{z})$ (Equation 21). To encourage coverage of all modes, given that we assume equal mixing weights for the full distribution, we introduce a separate loss term $\mathcal{L}_{uniform}$. For a batch of latent samples $\{\mathbf{z}^{(n)}\}_{n=1}^N$, we can define the average inverse distance to a component of the prior distribution as follows:

$$\bar{d}_k(\mathbf{z}) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{\|\mathbf{z}^{(n)} - \boldsymbol{\mu}_k\|^2}. \tag{26}$$

If the prior components are sufficiently far apart and we utilize equal mixing weights, the inverse distances $\bar{d}_k(\mathbf{z}) \in K$ approximate a uniform distribution for samples of prior $p(\mathbf{z})$. Consequently, we use this property to optimize our latent space to cover all models equally in expectation[5]:

$$\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\text{softmax}\left([\bar{d}_1(\mathbf{z}), \ldots, \bar{d}_K(\mathbf{z})]\right)\right] = \left[\frac{1}{K}, \ldots, \frac{1}{K}\right], \tag{27}$$

with short notation of the posterior distribution and its samples for readability. To optimize this expectation, we minimize the negative log-likelihood through a cross-entropy loss term:

$$\mathcal{L}_{uniform}(\mathbf{z}) = -\sum_{k=1}^{K}\frac{1}{K}\log\left(\text{softmax}\left([\bar{d}_1(\mathbf{z}), \ldots, \bar{d}_K(\mathbf{z})]\right)_k\right). \tag{28}$$

For the disruption risk variable, we create labels y^{t_m} using the time of disruption t_D as computed in [26]. If a discharge did not disrupt, it is set to 0 for each timestep. For discharges that disrupted, we define y^{t_m} to be 1 shortly before t_D with a linear ramp leading up to it as follows:

$$y^{t_m} = \begin{cases} 1, & \text{if } t_m \geq t_D - A \\ \frac{(t_m - (t_D - B))}{(B-A)}, & \text{if } t_D - B \leq t_m < t_D - A \\ 0, & \text{otherwise,} \end{cases} \tag{29}$$

for hyperparameters A and B. Then, we optimize the disruption risk using binary cross-entropy:

$$\begin{aligned}\mathcal{L}_{risk}(\mathbf{z}) = -\big[\, y^{t_m}\log(D_{risk}(\mathbf{z}^{t_m})) \\ + (1 - y^{t_m})\log(1 - D_{risk}(\mathbf{z}^{t_m}))\,\big].\end{aligned} \tag{30}$$

The total loss of the model is the sum of all components:

$$\mathcal{L} = a\mathcal{L}_{rec} + b\mathcal{L}_{KL} + c\mathcal{L}_{uniform} + d\mathcal{L}_{risk}, \tag{31}$$

for loss weights a, b, c and d. We train using minibatches of sequences from different shots. The sequences are split into the respective strided timewindows, and iteratively parsed by the model components. All parameters are jointly optimized with gradient-based optimization, that is, we apply backpropagation through time [57]. Remaining details are provided in Appendix A. An overview of the training and inference setup is given in Figure 4.

4.4. Architecture

Finally, we summarize the implementation of neural network functions f_ϕ, f_θ and f_D. For the encoder $f_\phi(\boldsymbol{\mu}^{t_{m-s}}, \mathbf{x}^{t_{m-w}:t_m})$ we utilize the Fourier Neural Operator (FNO) [58] to extract temporal information from timewindow input $\mathbf{x}^{t_{m-w}:t_m}$, as it has shown strong performance on modeling various fusion-related time series [32, 59–62]. The FNO performs nonlinear, global convolutions by applying linear global transformations in the frequency domain and nonlinear local transformations in the temporal domain. An FNO layer can be denoted as follows:

$$FNO^i : \mathbf{h}^i = \psi\big(\text{FFT}^{-1}(\mathbf{R}^i\text{FFT}(\mathbf{h}^{i-1})) + \mathbf{W}^i\mathbf{h}^{i-1}\big), \tag{32}$$

which maps an input signal $\mathbf{h}^{i-1}$ to its output $\mathbf{h}^i$. $\mathbf{R}^i \in \mathbb{R}^{D\times D\times M}$ and $\mathbf{W}^i \in \mathbb{R}^{D\times D}$ (D hidden dimensions; M fourier modes) are the learned parameters, ψ denotes the nonlinear activation function, and FFT denotes the Fast Fourier Transform [63].

The output of the FNO layers is flattened and concatenated to the previous value of $\boldsymbol{\mu}$, which is subsequently put through a small Multi-Layer Perceptron (MLP) [64]. We can summarize the encoder as:

$$f_\phi(\cdot) = MLP_\phi(FNO_\phi(\mathbf{x}^{t_{m-w}:t_m}), \boldsymbol{\mu}^{t_{m-s}}). \tag{33}$$

The decoder f_θ and disruption risk f_D follow a static formulation, and are implemented as MLPs. However, the low-frequency bias of neural networks [65] could limit the

[5] One can draw a parallel between this formulation and learning latent variables that are uninformative w.r.t. a property of choice, e.g. [56].

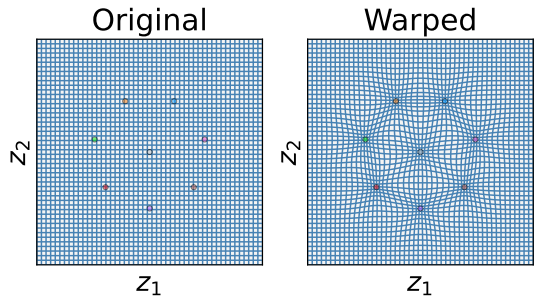

Figure 5: Precomputed deformation of a uniform space, used to ease the task of placing appropriate probability density on the prior modes during model training.

expressivity of the learned latent variable. To mitigate this issue, we first apply positional encoding to the latent variable before applying an MLP. This approach has been shown to aid in representing higher-frequency information [66], helping us better shape the latent space. The applied frequency encoding can be denoted as follows:

$$\gamma(p) = \big[\sin(2^0\pi p), \cos(2^0\pi p), \ldots, \sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p)\big], \tag{34}$$

for applying L frequencies to parameter p. We apply it separately to both dimensions. With $(z_1^{t_m}, z_2^{t_m}) = \mathbf{z}^{\mathbf{t_m}}$ denoting the two latent dimensions, we can summarize the decoder and disruption map as:

$$f_\theta(\cdot) = \mathit{MLP}_\theta(\gamma(z_1^{t_m}), \gamma(z_2^{t_m})), \tag{35}$$

$$f_D(\cdot) = \mathit{MLP}_D(\gamma(z_1^{t_m}), \gamma(z_2^{t_m})). \tag{36}$$

Additionally, to aid the optimization w.r.t. matching the prior[6], we add a precomputed warping of the latent space using the prior's parameters. Using mixture components $(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k)$, we place more density near the modes at the start of the optimization procedure, see Figure 5 for a depiction. An inverse of this transformation is applied on decoding from the latent space, ensuring it has no adverse effects on the decoder or disruption risk expressivity.

Finally, in order to better approximate the disruption rate using the disruption risk variable D_{risk}, we calibrate the variable post hoc. That is, after the model is trained, we can compute the actual disruption rate of all points in the learned latent space by projecting the dataset and evaluating the proportion of discharges that eventually disrupt for each location in $\mathbf{z}$. To ensure a continuous map, the projection is performed using the latent trajectories' distributions, that is, we sum the probability density functions of the inferred latent parameters. Then, we apply Platt scaling [67] to calibrate the prediction curve of D_{risk} to best represent the disruption rate.

[6]We found that training could lead to degenerate solutions where several prior modes were 'ignored'. A precomputed warping towards these modes results in a higher likelihood of placing some samples on a new peak, providing more informative gradients w.r.t. unexplored territories of the latent space.

5. Experiments and Results

In this section we evaluate the proposed method for learning a model that projects discharge measurements onto a lower-dimensional, interpretable latent variable. This latent variable is optimized to represent the measurements in an informative manner w.r.t. operational limits. Consequently, we evaluate (1) latent space properties in the context of disruption metrics; (2) the ability to separate distinct types of operational limits; and (3) using the model to identify patterns that can facilitate large-scale analyses. We assess (3) by doing a demonstrative study on parameters correlated with different types of disruptions.

A summary of the training details and model hyperparameters is provided in Section 5.1. In Section 5.2 we provide an analysis of the identified latent space and its relation to disruption metrics. Section 5.3 evaluates the ability to distinguish different types of disruptions and expands on the interpretability of the latent space w.r.t. physics quantities. Finally, Section 5.4 evaluates the utility of the model for downstream tasks, demonstrated with a proof-of-concept automatic analysis of disruption causes.

5.1. Dataset split and hyperparameters

Dataset split. The dataset (1629 shots) is split into a training set (1300 shots), validation set (165 shots) and test set (164 shots). The test and validation set are selected to be representative of the overall distribution while still being dissimilar to the train data. To select validation and test shots, we first compute a distance matrix of all shots using average values of operational parameters I_p, B_0, q_{95}, l_i, $n_{e,\text{core}}$, β_N, W_{tot}, P_{in} and P_{rad} (see also [32] for additional definitions) and subsequently find 10 clusters using agglomerative hierarchical clustering [68]. Then, we construct the test and validation set using a diversity maximization approach [69]. We sample from each cluster in proportion to its size in the full dataset. Specifically, we iteratively select the most distant shots to first construct the test set, and subsequently the validation set. As a result, we reserve a set of shots for the test and validation set that are both distant to the train shots while still covering a diverse parameter space.

Nevertheless, most evaluations also use the training data, given that we explore the identified latent variable $\mathbf{z}$ and its correspondence to disruption-related metadata not used during training; it is largely an unsupervised problem. The validation set is used during training to detect overfitting, whereas the test set is used to quantitatively evaluate disruption risk metrics on unseen data.

Hyperparameters. The main hyperparameters consider the dataset-related parameters and the prior structure. Remaining details on training and the model

architecture are found in Appendix A. The models are implemented using PyTorch [70], and we use net:cal [71] for post hoc Platt scaling.

For data-related parameters, we use timewindows of $w = 50$ timesteps and an equivalent stride of $s = 50$ timesteps (Equation 12). Given a data sampling rate of $10\,\mathrm{kHz}$, we generate latent trajectories with a timestep of $5\,\mathrm{ms}$. The disruption labels are computed using a ramp starting at $B = 1\,\mathrm{s}$ before t_D up to $A = 0.15\,\mathrm{s}$ before the time of disruption (Equation 29). A is similar to the current redistribution time on TCV [29], whereas B is selected heuristically to cover most of the flat-top dynamics, see Appendix A for more discussion. Additionally, all signals are standardized by subtracting the mean and dividing by the standard deviation using statistics computed on the train set.

For the prior structure (Equation 20), we scan a set of configurations that place one peak in the center surrounded by a set of equally spaced modes around this center. To quantify the benefit of adding more modes, we compute the mutual information [72] between the states and disruption-related event detections [26, 32]. At 7 surrounding peaks, for a total of $K = 8$ Gaussians in the mixture prior, the benefit of adding more modes levels off, see also Figure A.1. Consequently, we use this configuration as prior distribution.

5.2. Latent space and disruption metrics

Latent space density. Since we selected a dimensionality of 2 for latent variable $\mathbf{z}$, we can visualize the distribution in its entirety. We start with evaluating the spread around the latent space. In Figure 6 (left) we plot the probability density of the selected prior with $K = 8$ peaks, and label these peaks for the evaluation w.r.t. clusters later in this section. In Figure 6 (right) we see the learned distribution given our dataset. The latent distribution still follows a multimodal structure as desired, with deviations from the prior because of competing optimization objectives. Note that since the variable is learned, the axes are arbitrary up to a scaling of the model settings. Consequently, for all projections on $\mathbf{z}$ in this section we omit axis labels but rather keep the domain fixed.

Disruption risk. The identified disruption risk variable D_{risk} is depicted in Figure 7. The primary purpose of D_{risk} is as regularization that shapes the latent space to separate disruptive and nondisruptive regions. To validate that it can successfully provide such a separation, we fit a re-scaling of the 'raw' model output using Platt scaling w.r.t. the disruption rate and evaluate the difference[7]. We define this disruption rate as, for each location in $\mathbf{z}$, the fraction of timeslices projected there where the discharge ends in a disruption. We denote the real disruption rate as $\widetilde{D}_{rate}$ and

[7]The model did not have labels that represent the disruption rate during training. Consequently, directly comparing the raw model output to a calibrated quantity does not capture whether it represents the same information, rather, we fit a simple re-scaling to the model outputs first.

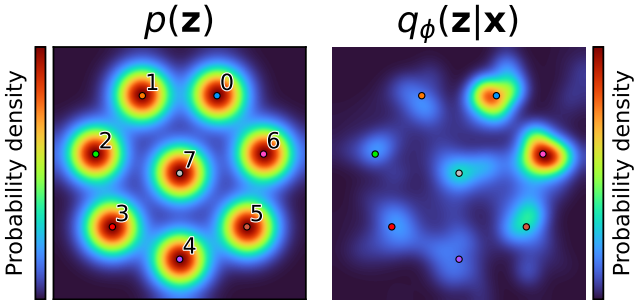

Figure 6: The probability density of the chosen prior distribution $p(\mathbf{z})$ and the learned posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$. Peaks of the prior, corresponding to cluster mapping $\mathcal{C}$, are labeled 0-7 (left). The learned latent variable $q_\phi(\mathbf{z}|\mathbf{x})$ shows a clear multimodal structure as desired (right), with deviations from the prior due to the the competing objectives in the joint model optimization.

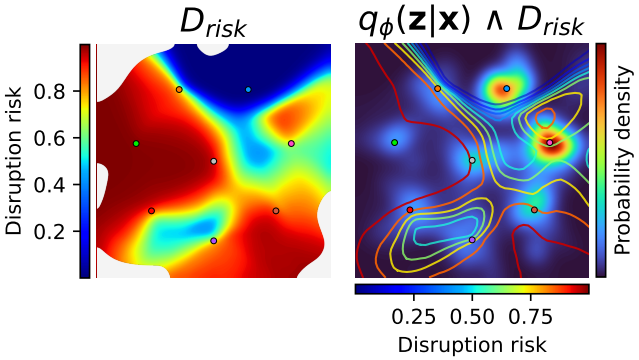

Figure 7: The learned disruption risk variable D_{risk} (left), overlaid on the posterior distribution (right). Regions with low and high disruption risk are spread throughout the latent space, with a zone of risk-free plasmas projected on the blue peak in the top right.

describe the deviation to D_{risk} using the expected calibration error (ECE) [73, 74]:

$$ECE = \sum_{m=1}^{M} \frac{N_m}{N} |\widetilde{D}_{rate}(\mathbf{z}_{\mathbf{B}_m}) - D_{risk}(\mathbf{z}_{\mathbf{B}_m})|, \qquad (37)$$

with N denoting the total number of samples. The ECE splits output range 0-1 into M bins of N_m samples each: the m^{th} bin covers rate $(\frac{m-1}{M}, \frac{m}{M}]$. The samples falling in the respective bin are denoted with $\mathbf{z}_{\mathbf{B}_m}$, and the ECE consequently measures the deviation of the predicted rate to the real rate. We can interpret the ECE as the average error between D_{risk} and the real disruption rate.

The ECE and the accompanying reliability diagram—a visualization of the ECE—are provided in Figure 8. D_{risk} accurately captures the real disruption rate, even when projecting a set of new, dissimilar shots (test set; bottom). Notably, there is a clear separation in the distribution of

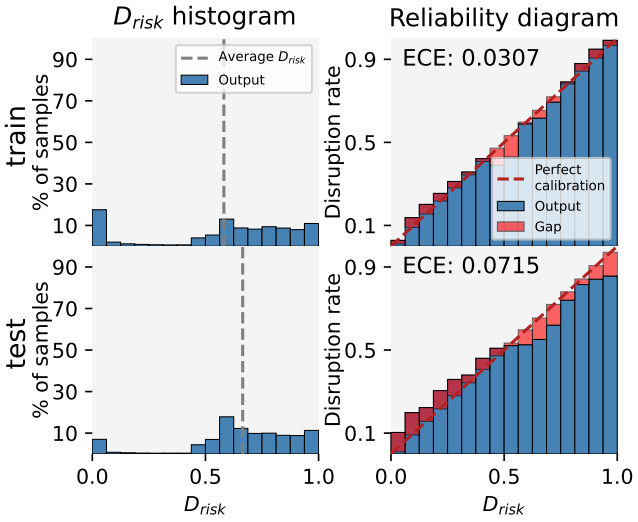

Figure 8: The distribution (left) and reliability diagrams (right) of D_{risk} w.r.t. the disruption rate. The reliability diagram provides a visual depiction of the expected calibration error (see Equation 37), with the corresponding metric values plotted on top. We define the disruption rate as, for each location in $\mathbf{z}$, the fraction of timeslices projected there where the discharge ends in a disruption. The value of D_{risk} is spread, with a notable lack of estimates between $\frac{1}{16}$ to $\frac{1}{2}$. It represents the actual disruption rate well, deviating only $\approx$3% on average for shots used to identify the latent space (top), and $\approx$7% for a set of novel discharges (bottom).

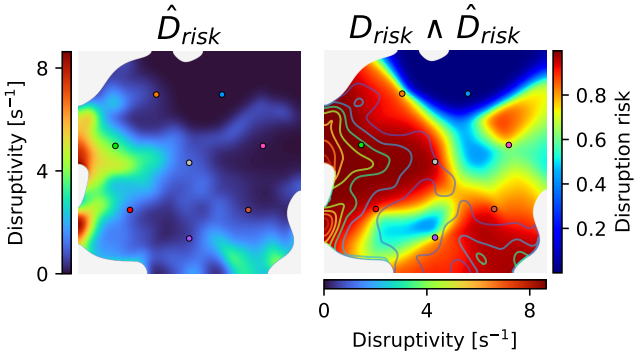

Figure 9: A projection of the disruptivity $\hat{D}_{disr}$ (left), overlaid on the disruption risk D_{risk} (right). We define $\hat{D}_{disr}$ as the number of disruptions per second for a given plasma property space. It is computed by summing the last projections of the disrupting discharges (0 to 5 ms before t_D) and dividing by posterior $q_\phi(\mathbf{z}|\mathbf{x})$, rescaled to the number of disruptions and the time span of the projections, respectively.

D_{risk} in the latent space: there are few locations with values between $\frac{1}{16}$ to $\frac{1}{2}$, rather, there is a region with almost no discharges ending in a disruption and a uniform spread for rates of $\approx\frac{1}{2}$ and higher.

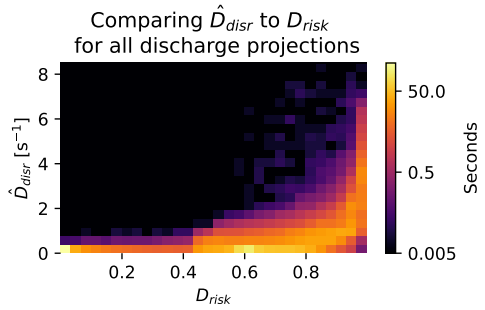

Figure 10: A comparison of D_{risk} and $\hat{D}_{disr}$. For low estimates of disruption risk (≤ 0.45), we find near-0 rates of disruptivity. That is, plasma regimes with low estimates for D_{risk} are distant from the actual onset of disruptions. As the risk increases, the disruptivity increases accordingly, with an exponential-like curve.

Disruptivity. Next, we compare D_{risk} to the disruptivity $\hat{D}_{\text{disr}}$, defined as the number of disruptions per second for a given plasma property space [7, 10]. We can compute this quantity in a continuous fashion by computing the distribution of projections just before t_D and dividing by the posterior, rescaled to the number of disruptions and the number of seconds of plasma dynamics, respectively. Figure 9 depicts $\hat{D}_{\text{disr}}$ and its comparison to D_{risk}. Expectedly, they significantly overlap, with higher values of disruptivity in zones of values for D_{risk} approaching 1. Low values of D_{risk} still contain many discharges with regular terminations, consequently, they have not crossed an 'uncontrollability' boundary. In Figure 10 we plot the histogram comparing both quantities, better quantifying this notion. At low values of D_{risk}, up to $\approx$0.45, we see virtually no disruptions, with an exponential curve as the risk increases.

Individual states. Finally, we analyze correlations between the individual states, i.e., the peaks in the posterior distribution, and known plasma quantities. To do so, we compute the fraction of time an event is detected when a plasma is considered to be in a given state as defined by $\mathcal{C}$ (Equation A.2). We first consider the plasma confinement state, automatically labeled using [32]. An overview of the fraction of time spent in each confinement state, for each state in $\mathbf{z}$, is provided in Figure 11. Note that we only consider detections with $\geq$75% confidence to ensure high-quality labels [32], consequently, the detections do not necessarily sum to 1 for each state. We see a clear separation, with states 0, 5, 6 and 7 containing (almost) only L-mode detections, and mostly H-mode detections in the other states. Even when using no information about the confinement state for training, the latent space can distinguish these regimes. Additionally, we compute the correlations with disruption-related event detections from [26]. We find less clear patterns compared to the confinement states, with the exception of neoclassical tearing modes (NTMs); see also Figure B.1.

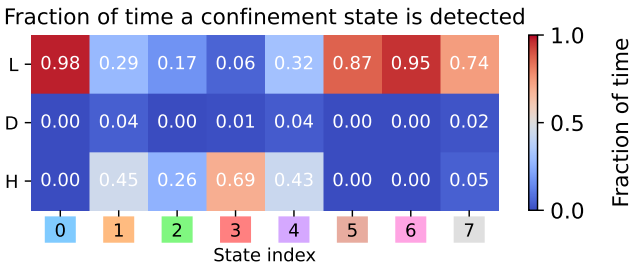

Figure 11: Correlation between high-confidence (≥ 0.75) confinement state detections computed using [32] and the states found in $\mathbf{z}$. For each state we denote the fraction of time L, D or H-mode is detected for the total time spent in a state. Even though no confinement state labels are used in this work, some separation is recovered by virtue of clustering the operating regimes.

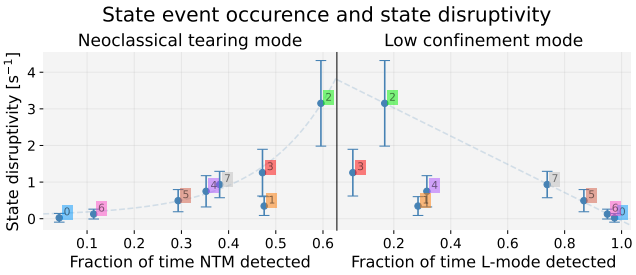

Figure 12: Comparing the two most commonly detected events, NTMs and L-mode, to the mean disruptivity in each state ($\pm$ standard deviation). We see clear patterns, with NTM occurence being associated with more disruptions, and conversely L-mode detections correlating with less disruptivity.

Most likely, many disruption-related events operate on much smaller timescales given the potentially very fast disruption dynamics in TCV [26–28], whereas $\mathbf{z}$ represents dynamics on larger timescales connected to global operating regimes.

Finally, we can correlate these events and their connection to the states' average disruptivity scores. For the two most commonly occuring detections, NTMs and L-mode, we plot this relation, see Figure 12. There is a clear correlation between the respective event occurrence and average disruptivity in each state, validiting the sensibility of the identifies states. For example, NTMs are often observed in connection to disruptions in TCV [29], and similarly L-mode plasmas generally operate further away from operational boundaries [10].

5.3. Distinguishing different types of disruptions

Disruption clustering. For evaluating the clustering of disruptions, we select a set of ≈200 disrupting shots that correspond to either the ITER Baseline (IBL) scenario [29], density limit (DL) experiments [30], or negative triangularity

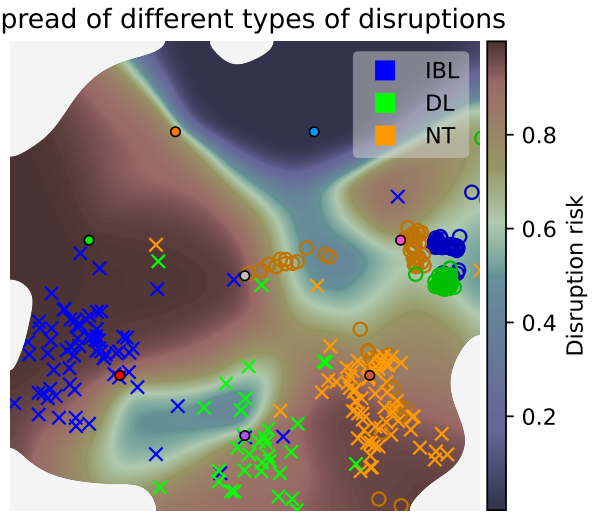

Figure 13: Projections at the start of the flat top (circle) and just before the time of disruption (cross), for a set of ≈200 discharges corresponding to the ITER Baseline (IBL), density limit (DL) experiments and negative triangularity (NT) configurations. At the start, the discharges are clustered together in a region with low disruption risk. At t_D, they are split onto different peaks in regions of higher D_{risk}.

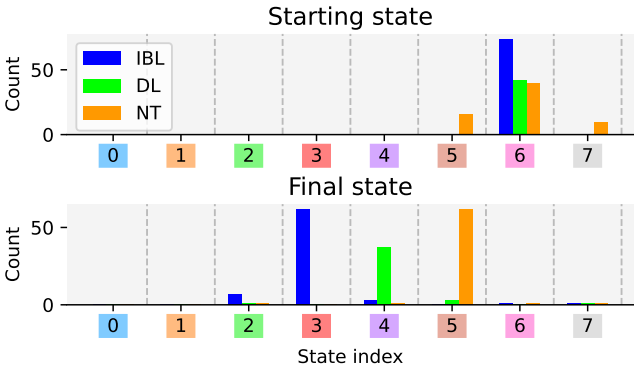

Figure 14: The assignment of the projections from Figure 13 to the different states using $\mathcal{C}$ (Equation A.2). While clustered together at the start of the flat top, there is a clear separation in the terminal states.

(NT) configurations [31]. For all these shots, we plot the projections at the start of the flat top (circle) and the last projection before the disruption[8] (cross) in Figure 13. The initial states are clustered, whereas there is a clear distinction at the time of disruption, see also Figure 14 for the state assignment at the start and end of the latent trajectories. Additionally, we see that initial states are projected in regions with lower values for D_{risk}, whereas there is a clear increase at the time of disruption. As such, D_{risk} seems to provide sensible correlations with known disruption proxies

[8] 0 to 5 ms up to t_D, depending on the alignment of the sampling rate and t_D.

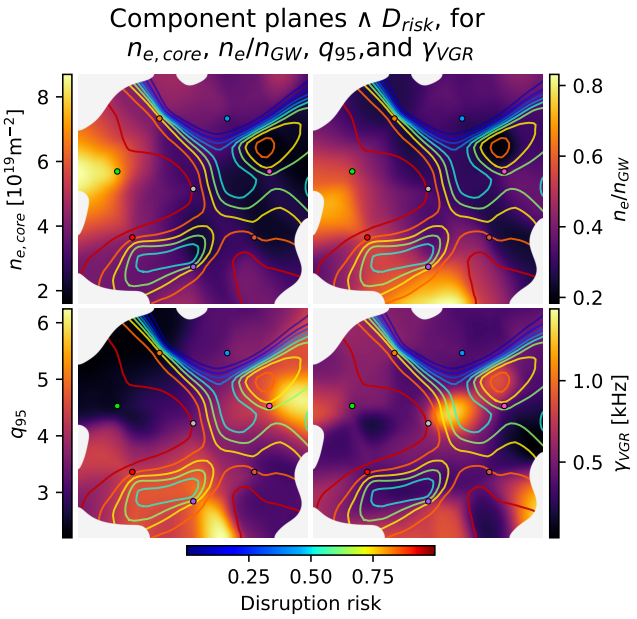

Figure 15: Component planes (projected using $p_\theta(\mathbf{x}|\mathbf{z})$) with the disruption risk overlaid on top. By projecting back to data space we can interpret the correlations with disruptive regimes. For example, we see a low q_{95} in regions of high D_{risk}, also corresponding to the region of most IBL-related disruptions. Similarly, we find a high Greenwald fraction near the DL-related disruptions, and an elevated vertical growth rate near the NT-related disruptions.

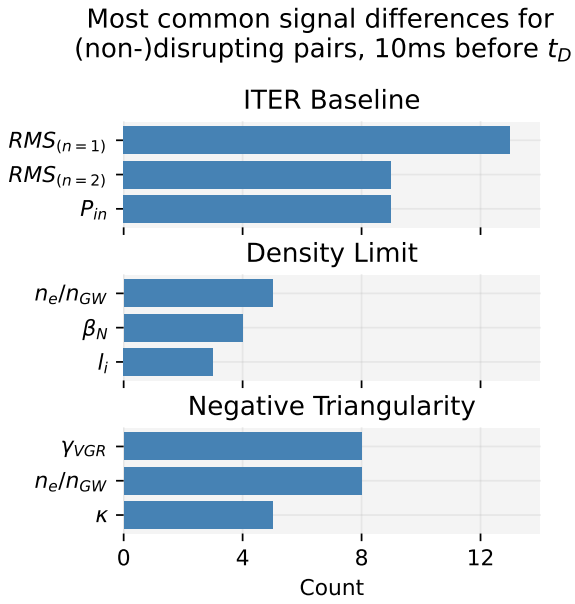

Figure 16: Automatically identified parameters correlating to disruptions. Parameters are identified by finding differences in data distributions just before $t_D - 10\,\mathrm{ms}$ (Figure 17) for a disrupting discharge and the most similar timestep for a counterfactual non-disrupting match. The identified features generally fall in line with past studies, i.e., we find MHD-related instabilities in the IBL case [29], recover the Greenwald fraction as most important parameter for DL disruptions [30] and find features associated with vertical instability for the NT scenarios [27, 31].

globally (Section 5.2), and locally different operational limits correspond to different regions.

Component planes. To better understand the clustering we can visualize the physics quantities as a function of $\mathbf{z}$ using decoder distribution $p_\theta(\mathbf{x}|\mathbf{z})$. These component planes, overlaid with the disruption risk, are depicted in Figure 15. We see correlations with the expected operational limit or characteristic quantities for the distinct categories of disruptions. For example, around the region of most IBL disruptions we find low values for q_{95} (≤ 4), a peak in the Greenwald fraction for the DL disruptions (≥ 0.7), and an elevated γ_{VGR} for the NT experiments, as a proxy for vertically unstable plasmas [27, 31].

5.4. Disruption counterfactual analysis

Counterfactual analysis. To be useful for downstream tasks, the latent space should be suitable for identifying non-trivial connections between different discharges. To demonstrate this principle, we conduct a demonstrative study on identifying features connected to disruptions. Specifically, we use the model to identify pairs of discharges that are similar, but one ends in a disruption whereas the other ends in a regular termination. If the two are sufficiently similar otherwise, they can be used to do (approximate) counterfactual analysis [75]. We additionally use the model to identify the most similar timestep in the non-disrupting shot, compared to $10\,\mathrm{ms}$ before t_D for the disrupting case. Consequently, we can automatically identify parameters connected with disruptions in an interpretable manner.

Procedure. The low dimensionality of $\mathbf{z}$ allows us to efficiently compute distances between shots using Dynamic Time Warping (DTW) [76]. For each scenario and its disrupting discharges, we compute the optimal alignment w.r.t. all other discharges of the same category, and select the closest discharge as the counterfactual case. Then, we compute the closest point in the latent space between the disrupting shot $10\,\mathrm{ms}$ before disruption and a timestep in the counterfactual shot, and sample parameters for both discharges using a half-Gaussian distribution with $\sigma = 10\,\mathrm{ms}$ at this point. See Figure 17 for an example of an automatically identified pair and its corresponding sampling window, along with their projections and timestep-of-interest in Figure 18.

By computing a distance metric on a normalized scale—standardizing feature values using the train set statistics—we can compare the relative differences in feature values. To quantify this difference, we utilize the Wasserstein distance [77], which can be interpreted as a cost of transforming one distribution to the other. We plot the most

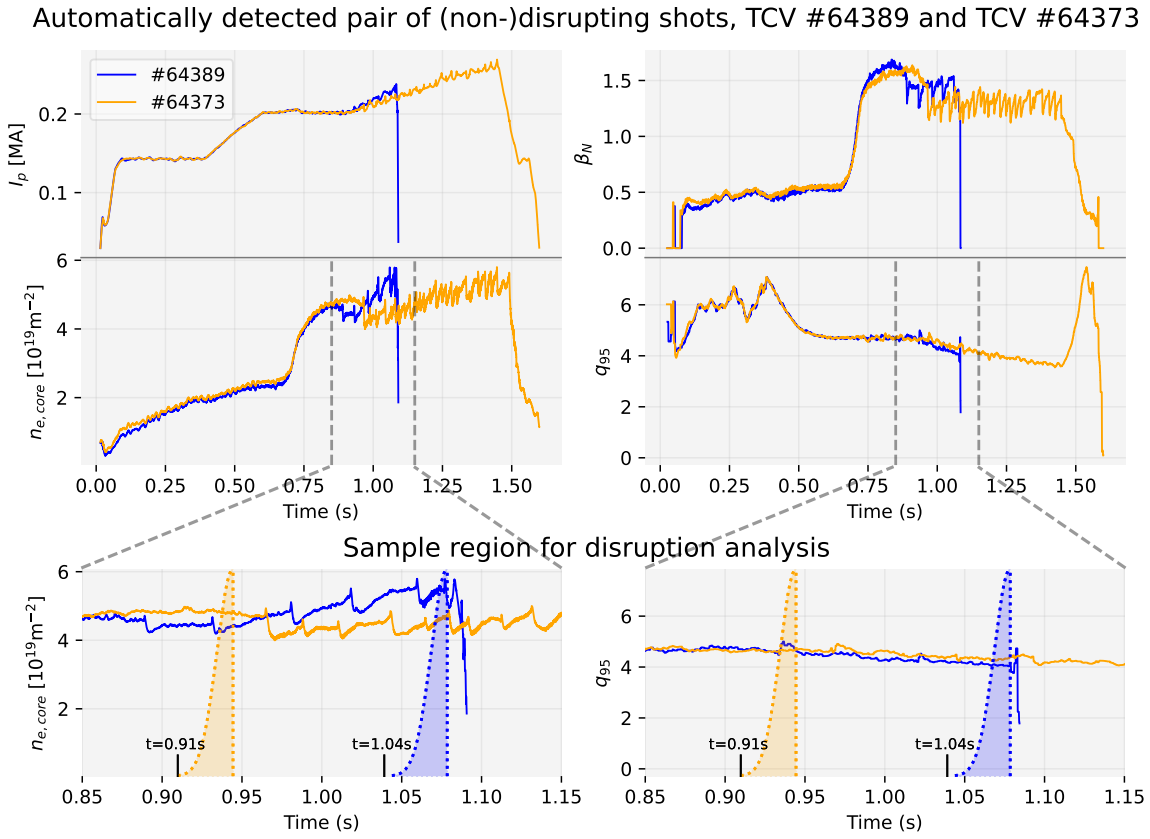

Figure 17: An example discharge from IBL scenario development experiments that ended in a disruption, TCV #64389, and its automatically identified counterfactual that did not disrupt, TCV #64373 (top). We find the closest point in the latent space (Figure 18) for #64389 compared to #64373 10 ms before t_D, and sample signal values in this window to identify significant differences between the two shots (bottom).

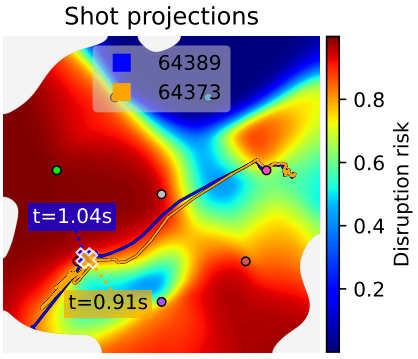

Figure 18: Projections of TCV #64389 and TCV #64373 in the latent space (on top of D_{risk}). We mark the position 10 ms before t_D for #64389 and the corresponding comparison point used for #64373.

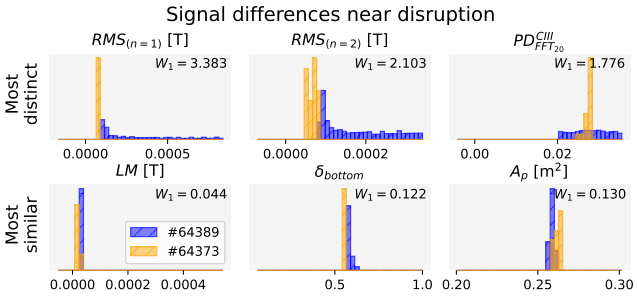

Figure 19: Empirical distributions up to 10 ms before t_D for the disrupting case and the identified closest point for the counterfactual case. We order the signals by 1-Wasserstein (W_1) distance on standardized feature values, and plot the top 3 most dissimilar and the top 3 most similar features.

distinct and least distinct empirical feature distributions for the example case in Figure 19. By placing a minimum threshold on the DTW distance between the shots and the Wasserstein difference between the signal distributions, we can robustly identify relevant features on a larger scale. Specifically, we select shots with a *DTW* distance of at most 100 in latent-space scale (for reference, the visualized domain of $\mathbf{z}$ spans [-2.2, 2.2]), and a standardized Wasserstein distance $W_1 \geq 3$.

Results. We plot the most commonly identified significant feature difference for all IBL, DL and NT disruptions in Figure 16. The identified features generally align with known limits. The IBL scenario performance is often limited by MHD instabilities on TCV [29], and consequently discrepancies in MHD activity are commonly the difference between (non-)disrupting shots. For the density limit, we expect the Greenwald fraction as the primary operational limit [30]. Lastly, negative triangularity on TCV is known to be vertically unstable [27, 31], which is well captured by the vertical growth rate γ_{VGR} signal.

Notably, by framing the automated analysis as identifying appropriate discharges and times therein, we can include additional signals not used in the model and compare different signals at arbitrary sampling rates, highlighting the flexibility of the approach.

6. Conclusions and Discussion

We have presented a method that identifies a low-dimensional, abstract representation of the operational space of a tokamak. The method extends the VAE framework for properties desired for plasma state monitoring and disruption characterization. Specifically, the representation is optimized to model time series of discharge measurements as latent trajectories, to provide a clustering for distinct operating regimes, all while being informative w.r.t. the disruption risk.

We have evaluated the method using a dataset of approximately 1600 TCV discharges modeling dynamics in the flat-top phase, covering flat-top disruptions and regularly-terminating shots. We investigated the identified latent variable w.r.t. the identified disruption-risk variable and its correspondence to the actual disruption rate and disruptivity computed after training. Additionally, we compared the identified states with known plasma state descriptions from external tools [26, 32]. We validated the ability to separate disruptions associated with distinct operational limits, as found in ITER Baseline scenarios, density limit experiments and negative triangularity configurations. Finally, we demonstrated the utility of the tool for downstream analyses by conducting an exploratory study on disruption-related features using counterfactual analysis.

The main downside of the approach is the relatively slow timescales that are represented in the latent space. The limited capacity of a low-dimensional latent variable coupled with the large variety in plasma scenarios means only a limited amount of information can be represented. However, a particular focus on the fast timescales around the time of disruption is of interest for analysis regarding the associated chain-of-events.

6.1. Future work

Modeling fast timescales requires an increased expressivity of the generative model. Likely, a trade-off has to be made between interpretability and expressivity, given that the 2D latent variable forms one of the main bottlenecks in this regard. Potential approaches include increasing the latent space dimensionality, extending the latent variable to a hierarchical structure [78] or utilizing alternative latent variable formulations, e.g. based on normalizing flows [79].

To extend towards multi-machine analysis, it is of interest to learn a single latent representation found through discharges coming from different tokamaks. Such an approach comes with the challenge of properly integrating dynamics that operate on different time and spatial scales—

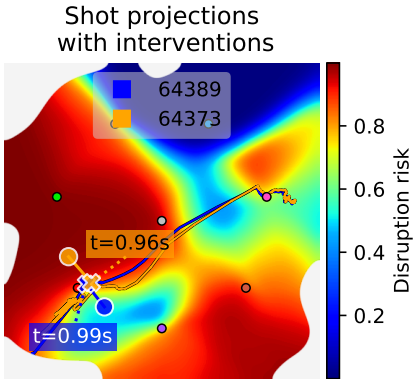

Figure 20: Alteration of discharge projections just before the disruption towards a safer region (#64389), or later in the discharge (#64373) towards a less safe region.

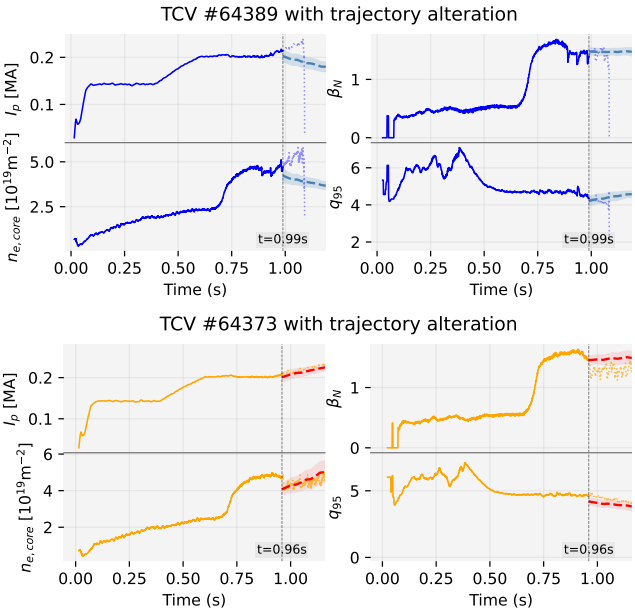

Figure 21: Projections of the altered trajectories illustrated in Figure 20 in data space. Potentially, one could utilize a latent variable approach to project ongoing discharges in real time, to identify parameter changes gradually steering a discharge towards less disruption-prone regimes.

a latent variable that simply clusters each device into a separate region provides little benefit over learning a representation on a per-device basis. Notably, prior works have studied multi-machine disruption prediction [14–16] and multi-machine representation learning in non-disruption contexts [80], providing a basis for this extension.

Finally, the latent space-approach could be utilized in the setting of plasma control. For example, one could project a discharge onto $\mathbf{z}$ in real time as new measurements come in, and explore the surrounding region to inform control targets. One could project in the direction of a lower disruption risk to find physics quantities close to the current regime that are connected to less disruption risk. See Figure 20 as an example of such an intervention for #64389, along with Figure 21 for the corresponding quantities in data-space. Or to the contrary, one could project towards regions of higher disruptivity to inform a controller of parameter spaces to avoid, see #64373 in Figures 20 and 21. Additionally, one could extend the formulation to include a forward model that predicts future states based on control actions [81], providing even more information for advanced control schemes [47, 82].

Acknowledgements

This work was funded in part by a Swiss Data Science Center project grant (C21-14). This work has been carried out within the framework of the EUROfusion Consortium, partially funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). The Swiss contribution to this work has been funded in part by the Swiss State Secretariat for Education, Research and Innovation (SERI). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union, the European Commission or SERI. Neither the European Union nor the European Commission nor SERI can be held responsible for them. This work was supported in part by the Swiss National Science Foundation. This work used the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. EINF-7709.

References

[1] Jepu, I., Matthews, G., Widdowson, A., Rubel, M., Fortuna-Zaleśna, E., Zdunek, J., Petersson, P., Thompson, V., Dinca, P. and Porosnicu, C. et al. Beryllium melting and erosion on the upper dump plates in JET during three ITER-like wall campaigns. *Nuclear Fusion*, 59(8):086009, jun 2019. https://dx.doi.org/10.1088/1741-4326/ab2076, doi:10.1088/1741-4326/ab2076.

[2] Jepu, I., Widdowson, A., Matthews, G., Coad, J., Likonen, J., Brezinsek, S., Rubel, M., Pintsuk, G., Petersson, P. and Fortuna-Zalesna, E. et al. Overview of damage to beryllium limiters by unmitigated disruptions and runaway electrons in the JET tokamak with metal walls. *Nuclear Fusion*, 64(10):106047, sep 2024. https://dx.doi.org/10.1088/1741-4326/ad6614, doi:10.1088/1741-4326/ad6614.

[3] Lehnen, M., Aleynikova, K., Aleynikov, P., Campbell, D., Drewelow, P., Eidietis, N., Gasparyan, Y., Granetz, R., Gribov, Y. and Hartmann, N. et al. Disruptions in ITER and strategies for their control and mitigation. *Journal of Nuclear Materials*, 463:39–48, 2015. https://www.sciencedirect.com/science/article/pii/S0022311514007594, doi:https://doi.org/10.1016/j.jnucmat.2014.10.075.

[4] Strait, E., Barr, J., Baruzzo, M., Berkery, J., Buttery, R., de Vries, P., Eidietis, N., Granetz, R., Hanson, J. and Holcomb, C. et al. Progress in disruption prevention for ITER. *Nuclear Fusion*, 59(11):112012, jun 2019. https://dx.doi.org/10.1088/1741-4326/ab15de, doi:10.1088/1741-4326/ab15de.

[5] Fasoli, A. Essay: Overcoming the obstacles to a magnetic fusion power plant. *Phys. Rev. Lett.*, 130:220001, May 2023. https://link.aps.org/doi/10.1103/PhysRevLett.130.220001, doi:10.1103/PhysRevLett.130.220001.

[6] Wesson, J. *Tokamaks*. International Series of Monographs on Physics. Clarendon Press, Oxford, England, 3 edition, November 2003. https://books.google.ch/books?id=iPlAwZI6HIYC.

[7] Bandyopadhyay, I., Igochine, V., Sauter, O., Sabbagh, S., Park, J.-K., Nardon, E., Villone, F., Maraschek, M., Pautasso, G. and Eidietis, N. et al. *3.3 Disruption Prediction*. Chapter 4: MHD, Disruptions and Control Physics in Tokamaks. Submitted to Nucl. Fusion.

[8] Hender, T., Wesley, J., Bialek, J., Bondeson, A., Boozer, A., Buttery, R., Garofalo, A., Goodman, T., Granetz, R. and Gribov, Y. et al. Chapter 3: MHD stability, operational limits and disruptions. *Nuclear Fusion*, 47(6):S128, jun 2007. https://dx.doi.org/10.1088/0029-5515/47/6/S03, doi:10.1088/0029-5515/47/6/S03.

[9] de Vries, P., Johnson, M., Segui, I., and JET EFDA Contributors. Statistical analysis of disruptions in JET. *Nuclear Fusion*, 49(5):055011, apr 2009. https://dx.doi.org/10.1088/0029-5515/49/5/055011, doi:10.1088/0029-5515/49/5/055011.

[10] de Vries, P., Johnson, M., Alper, B., Buratti, P., Hender, T., Koslowski, H., Riccardo, V., and JET-EFDA Contributors. Survey of disruption causes at JET. *Nuclear Fusion*, 51(5):053018, apr 2011. https://dx.doi.org/10.1088/0029-5515/51/5/053018, doi:10.1088/0029-5515/51/5/053018.

[11] Gerasimov, S., Abreu, P., Artaserse, G., Baruzzo, M., Buratti, P., Carvalho, I., Coffey, I., De La Luna, E., Hender, T. and Henriques, R. et al. Overview of disruptions with JET-ILW. *Nuclear Fusion*, 60(6):066028, may 2020. https://dx.doi.org/10.1088/1741-4326/ab87b0, doi:10.1088/1741-4326/ab87b0.

[12] LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521(7553):436–444, May 2015. doi:10.1038/nature14539.

[13] Rea, C., Montes, K., Erickson, K., Granetz, R., and Tinguely, R. A real-time machine learning-based disruption predictor in DIII-D. *Nuclear Fusion*, 59(9):096016, jul 2019. https://dx.doi.org/10.1088/1741-4326/ab28bf, doi:10.1088/1741-4326/ab28bf.

[14] Zheng, W., Xue, F., Chen, Z., Chen, D., Guo, B., Shen, C., Ai, X., Wang, N., Zhang, M. and Ding, Y. et al. Disruption prediction for future tokamaks using parameter-based transfer learning. *Communications Physics*, 6(1):181, Jul 2023. doi:10.1038/s42005-023-01296-9.

[15] Zhu, J., Rea, C., Montes, K., Granetz, R., Sweeney, R., and Tinguely, R. Hybrid deep-learning architecture for general disruption prediction across multiple tokamaks. *Nuclear Fusion*, 61(2):026007, dec 2020. https://dx.doi.org/10.1088/1741-4326/abc664, doi:10.1088/1741-4326/abc664.

[16] Kates-Harbeck, J., Svyatkovskiy, A., and Tang, W. Predicting disruptive instabilities in controlled fusion plasmas through deep learning. *Nature*, 568(7753):526–531, Apr 2019. doi:10.1038/s41586-019-1116-4.

[17] Vega, J., Murari, A., Dormido-Canto, S., Rattá, G. A., Gelfusa, M.,

and JET Contributors. Disruption prediction with artificial intelligence techniques in tokamak plasmas. *Nature Physics*, 18(7):741–750, Jul 2022. doi:10.1038/s41567-022-01602-2.

[18] Aymerich, E., Sias, G., Pisano, F., Cannas, B., Carcangiu, S., Sozzi, C., Stuart, C., Carvalho, P., Fanni, A., and JET Contributors. Disruption prediction at JET through deep convolutional neural networks using spatiotemporal information from plasma profiles. *Nuclear Fusion*, 62(6):066005, apr 2022. https://dx.doi.org/10.1088/1741-4326/ac525e, doi:10.1088/1741-4326/ac525e.

[19] Aledda, R., Cannas, B., Fanni, A., Pau, A., Sias, G., and the ASDEX Upgrade Team. Improvements in disruption prediction at ASDEX upgrade. *Fusion Engineering and Design*, 96-97:698–702, 2015. Proceedings of the 28th Symposium On Fusion Technology (SOFT-28). https://www.sciencedirect.com/science/article/pii/S0920379615002148, doi:https://doi.org/10.1016/j.fusengdes.2015.03.045.

[20] Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014. https://openreview.net/forum?id=33X9fd2-9FyZd.

[21] Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR. https://proceedings.mlr.press/v32/rezende14.html.

[22] Girin, L., Leglaive, S., Bie, X., Diard, J., Hueber, T., and Alameda-Pineda, X. Dynamical variational autoencoders: A comprehensive review. *Found. Trends Mach. Learn.*, 15(1–2):1–175, December 2021. doi:10.1561/2200000089.

[23] Dilokthanakul, N., Mediano, P. A. M., Garnelo, M., Lee, M. C. H., Salimbeni, H., Arulkumaran, K., and Shanahan, M. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2017. arXiv:1611.02648, doi:10.48550/arXiv.1611.02648.

[24] Tomczak, J. M. *Deep Generative Modeling*. Springer International Publishing, 2022. http://dx.doi.org/10.1007/978-3-030-93158-2, doi:10.1007/978-3-030-93158-2.

[25] Duval, B., Abdolmaleki, A., Agostini, M., Ajay, C., Alberti, S., Alessi, E., Anastasiou, G., Andrèbe, Y., Apruzzese, G. and Auriemma, F. et al. Experimental research on the TCV tokamak. *Nuclear Fusion*, 64(11):112023, oct 2024. https://dx.doi.org/10.1088/1741-4326/ad8361, doi:10.1088/1741-4326/ad8361.

[26] Pau, A., Sauter, O., Sommariva, C., Poels, Y., Venturini, C., Labit, B., Imbeaux, F., Litaudon, X., Falchetto, G. and Joffrin, E. et al. A modern framework to support disruption studies: the EUROfusion disruption database. In *29th Fusion Energy Conference (IAEA-FEC)*, 2023. https://conferences.iaea.org/event/316/contributions/28183/.

[27] Marchioni, S. *Vertical Instability Studies in the TCV Tokamak and Development and Application of Multimachine Real-Time Proximity Control Strategies*. PhD thesis, EPFL, Lausanne, 2024. https://infoscience.epfl.ch/handle/20.500.14299/242254, doi:10.5075/epfl-thesis-10943.

[28] Turri, G., Sauter, O., Porte, L., Alberti, S., Asp, E., Goodman, T. P., Martin, Y. R., Udintsev, V. S., and Zucca, C. The role of MHD in the sustainment of electron internal transport barriers and H-mode in TCV. *Journal of Physics: Conference Series*, 123(1):012038, jul 2008. https://dx.doi.org/10.1088/1742-6596/123/1/012038, doi:10.1088/1742-6596/123/1/012038.

[29] Labit, B., Sauter, O., Pütterich, T., Bagnato, F., Camenen, Y., Coda, S., Contré, C., Coosemans, R., Eriksson, F. and Février, O. et al. Progress in the development of the ITER baseline scenario in TCV. *Plasma Physics and Controlled Fusion*, 66(2):025016, jan 2024. https://dx.doi.org/10.1088/1361-6587/ad1a40, doi:10.1088/1361-6587/ad1a40.

[30] Sieglin, B., Maraschek, M., Gude, A., Klossek, F., Felici, F., Bernert, M., Kudlacek, O., Pau, A., Piron, L. and Lennholm, M. et al. H-mode density limit disruption avoidance in ASDEX Upgrade, TCV and JET. *Fusion Engineering and Design*, 215:114961, 2025. https://www.sciencedirect.com/science/article/pii/S0920379625001619, doi:https://doi.org/10.1016/j.fusengdes.2025.114961.

[31] Coda, S., Merle, A., Sauter, O., Porte, L., Bagnato, F., Boedo, J., Bolzonella, T., Février, O., Labit, B. and Marinoni, A. et al. Enhanced confinement in diverted negative-triangularity L-mode plasmas in TCV. *Plasma Physics and Controlled Fusion*, 64(1):014004, dec 2021. https://dx.doi.org/10.1088/1361-6587/ac3fec, doi:10.1088/1361-6587/ac3fec.

[32] Poels, Y., Venturini, C., Pau, A., Sauter, O., Menkovski, V., the TCV team, and the WPTE team. Robust confinement state classification with uncertainty quantification through ensembled data-driven methods. *arXiv preprint arXiv:2502.17397*, 2025. arXiv:2502.17397, doi:10.48550/arXiv.2502.17397.

[33] Bishop, C. M., Svensén, M., and Williams, C. K. I. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998. doi:10.1162/089976698300017953.

[34] Pau, A., Fanni, A., Carcangiu, S., Cannas, B., Sias, G., Murari, A., Rimini, F., and the JET Contributors. A machine learning approach based on generative topographic mapping for disruption prevention and avoidance at JET. *Nuclear Fusion*, 59(10):106017, aug 2019. https://dx.doi.org/10.1088/1741-4326/ab2ea9, doi:10.1088/1741-4326/ab2ea9.

[35] Aymerich, E., Fanni, A., Sias, G., Carcangiu, S., Cannas, B., Murari, A., Pau, A., and JET contributors. A statistical approach for the automatic identification of the start of the chain of events leading to the disruptions at JET. *Nuclear Fusion*, 61(3):036013, feb 2021. https://dx.doi.org/10.1088/1741-4326/abcb28, doi:10.1088/1741-4326/abcb28.

[36] Kohonen, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, Jan 1982. doi:10.1007/BF00337288.

[37] Aledda, R., Cannas, B., Fanni, A., Sias, G., and Pautasso, G. Mapping of the ASDEX upgrade operational space for disruption prediction. *IEEE Transactions on Plasma Science*, 40(3):570–576, 2012. doi:10.1109/TPS.2011.2174385.

[38] Aymerich, E., Fanni, A., Pisano, F., Sias, G., Cannas, B., JET Contributors, and WPTE Team. A self-organised partition of the high dimensional plasma parameter space for plasma disruption prediction. *Nuclear Fusion*, 64(10):106063, sep 2024. https://dx.doi.org/10.1088/1741-4326/ad7474, doi:10.1088/1741-4326/ad7474.

[39] Wei, Y., Levesque, J., Hansen, C., Mauel, M., and Navratil, G. A dimensionality reduction algorithm for mapping tokamak operational regimes using a variational autoencoder (VAE) neural network. *Nuclear Fusion*, 61(12):126063, nov 2021. https://dx.doi.org/10.1088/1741-4326/ac3296, doi:10.1088/1741-4326/ac3296.

[40] Bürli, A., Pau, A., Koller, T., Sauter, O., and Contributors, J. Towards transparent and accurate plasma state monitoring at JET. *arXiv preprint arXiv:2502.12182*, 2025. arXiv:2502.12182, doi:10.48550/arXiv.2502.12182.

[41] Sabbagh, S. A., Berkery, J. W., Park, Y. S., Butt, J., Riquezes, J. D., Bak, J. G., Bell, R. E., Delgado-Aparicio, L., Gerhardt, S. P. and Ham, C. J. et al. Disruption event characterization and forecasting in tokamaks. *Physics of Plasmas*, 30(3):032506, 03 2023. https://pubs.aip.org/aip/pop/article-pdf/doi/10.1063/5.0133825/19824942/032506_1_online.pdf, doi:10.1063/5.0133825.

[42] Maraschek, M., Gude, A., Igochine, V., Zohm, H., Alessi, E., Bernert, M., Cianfarani, C., Coda, S., Duval, B. and Esposito, B. et al. Path-oriented early reaction to approaching disruptions in ASDEX Upgrade and TCV in view of the future needs for ITER and DEMO. *Plasma Physics and Controlled Fusion*, 60(1):014047, nov

2017. https://dx.doi.org/10.1088/1361-6587/aa8d05, doi:10.1088/1361-6587/aa8d05.

[43] Greenwald, M. Density limits in toroidal plasmas. *Plasma Physics and Controlled Fusion*, 44(8):R27, jul 2002. https://dx.doi.org/10.1088/0741-3335/44/8/201, doi:10.1088/0741-3335/44/8/201.

[44] Kruskal, M. and Tuck, J. L. The instability of a pinched fluid with a longitudinal magnetic field. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 245(1241):222–237, June 1958. http://dx.doi.org/10.1098/rspa.1958.0079, doi:10.1098/rspa.1958.0079.

[45] Shafranov, V. D. On magnetohydrodynamical equilibrium configurations. *Soviet Phys. JETP*, Vol: 6, 03 1958. https://www.osti.gov/biblio/4305963.

[46] Zohm, H. *Current Driven Ideal MHD Modes in a Tokamak*, chapter 4, pages 55–68. John Wiley & Sons, Ltd, 2014. https://onlinelibrary.wiley.com/doi/abs/10.1002/9783527677375.ch4, doi:https://doi.org/10.1002/9783527677375.ch4.

[47] Vu, T., Felici, F., Galperti, C., Maraschek, M., Pau, A., Rispoli, N., Sauter, O., and Sieglin, B. Integrated real-time supervisory management for off-normal-event handling and feedback control of tokamak plasmas. *IEEE Transactions on Nuclear Science*, 68(8):1855–1861, 2021. doi:10.1109/TNS.2021.3084410.

[48] Moret, J.-M., Duval, B., Le, H., Coda, S., Felici, F., and Reimerdes, H. Tokamak equilibrium reconstruction code LIUQE and its real time implementation. *Fusion Engineering and Design*, 91:1–15, 2015. https://www.sciencedirect.com/science/article/pii/S0920379614005973, doi:https://doi.org/10.1016/j.fusengdes.2014.09.019.

[49] Testa, D., Team, E. M., and Team, T. Manufacturing, installation, commissioning, and first results with the 3D low-temperature co-fired ceramic high-frequency magnetic sensors on the Tokamak à Configuration Variable. *Review of Scientific Instruments*, 91(8):081401, 08 2020. doi:10.1063/1.5115004.

[50] Kullback, S. and Leibler, R. A. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. http://www.jstor.org/stable/2236703.

[51] Leglaive, S., Alameda-Pineda, X., Girin, L., and Horaud, R. A recurrent variational autoencoder for speech enhancement. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 371–375, 2020. doi:10.1109/ICASSP40776.2020.9053164.

[52] Dang, H., Huu, T. T., Nguyen, T. M., and Ho, N. Beyond vanilla variational autoencoders: Detecting posterior collapse in conditional and hierarchical variational autoencoders. In *The Twelfth International Conference on Learning Representations*, 2024. https://openreview.net/forum?id=4zZFGliCl9.

[53] Chen, T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, pages 6572–6583, 2018. https://proceedings.neurips.cc/paper/2018/hash/69386f6bb1dfed68692a24c8686939b9-Abstract.html.

[54] Minka, T. et al. Divergence measures and message passing. Technical report, Technical report, Microsoft Research, 2005. https://miat.inrae.fr/AIGM/biblios/TR-2005-173.pdf.

[55] Jones, A. KL($q||p$) is mode-seeking. Accessed 2025-03-26. https://andrewcharlesjones.github.io/journal/klqp.html.

[56] Zheng, Z. and Sun, L. Disentangling latent space for VAE by label relevant/irrelevant dimensions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. https://openaccess.thecvf.com/content_CVPR_2019/html/Zheng_Disentangling_Latent_Space_for_VAE_by_Label_RelevantIrrelevant_Dimensions_CVPR_2019_paper.html.

[57] Werbos, P. J. Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, 1(4):339–356, 1988. https://www.sciencedirect.com/science/article/pii/089360808890007X, doi:https://doi.org/10.1016/0893-6080(88)90007-X.

[58] Li, Z., Kovachki, N. B., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A. M., and Anandkumar, A. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, volume 9, 2021. https://openreview.net/forum?id=c8P9NQVtmnO.

[59] Gopakumar, V., Pamela, S., Zanisi, L., Li, Z., Gray, A., Brennand, D., Bhatia, N., Stathopoulos, G., Kusner, M. and Deisenroth, M. P. et al. Plasma surrogate modelling using fourier neural operators. *Nuclear Fusion*, 64(5):056025, apr 2024. https://dx.doi.org/10.1088/1741-4326/ad313a, doi:10.1088/1741-4326/ad313a.

[60] Poels, Y., Derks, G., Westerhof, E., Minartz, K., Wiesen, S., and Menkovski, V. Fast dynamic 1D simulation of divertor plasmas with neural PDE surrogates. *Nuclear Fusion*, 63(12):126012, sep 2023. https://dx.doi.org/10.1088/1741-4326/acf70d, doi:10.1088/1741-4326/acf70d.

[61] Gopakumar, V., Gray, A., Zanisi, L., Nunn, T., Pamela, S., Giles, D., Kusner, M. J., and Deisenroth, M. P. Calibrated physics-informed uncertainty quantification. *arXiv preprint arXiv:2502.04406*, 2025. arXiv:2502.04406, doi:10.48550/arXiv.2502.04406.

[62] Pamela, S., Carey, N., Brandstetter, J., Akers, R., Zanisi, L., Buchanan, J., Gopakumar, V., Hoelzl, M., Huijsmans, G. and Pentland, K. et al. Neural-parareal: Self-improving acceleration of fusion mhd simulations using time-parallelisation and neural operators. *Computer Physics Communications*, 307:109391, 2025. https://www.sciencedirect.com/science/article/pii/S001046552400314X, doi:https://doi.org/10.1016/j.cpc.2024.109391.

[63] Cooley, J. W. and Tukey, J. W. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965. doi:10.1090/s0025-5718-1965-0178586-1.

[64] Ivakhnenko, A. G. Polynomial theory of complex systems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1(4):364–378, 1971. doi:10.1109/TSMC.1971.4308320.

[65] Xu, Z.-Q. J., Zhang, Y., and Xiao, Y. Training behavior of deep neural network in frequency domain. In Gedeon, T., Wong, K. W., and Lee, M., editors, *Neural Information Processing*, pages 264–274, Cham, 2019. Springer International Publishing. http://dx.doi.org/10.1007/978-3-030-36708-4_22, doi:10.1007/978-3-030-36708-4_22.

[66] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. NeRF: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, December 2021. doi:10.1145/3503250.

[67] Platt, J. et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999. https://www.bibsonomy.org/bibtex/1c5df9f9137085cad9cafce3c347b2508.

[68] Murtagh, F. and Contreras, P. Algorithms for hierarchical clustering: an overview. *WIREs Data Mining and Knowledge Discovery*, 2(1):86–97, 2012. https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.53, doi:https://doi.org/10.1002/widm.53.

[69] Chandra, B. and Halldórsson, M. M. Approximation algorithms for dispersion problems. *Journal of Algorithms*, 38(2):438–465, 2001. https://www.sciencedirect.com/science/article/pii/S0196677400911453, doi:https://doi.org/10.1006/jagm.2000.1145.

[70] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N. and Antiga, L. et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019. https:

//proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

[71] Kuppers, F., Kronenberger, J., Shantia, A., and Haselhoff, A. Multivariate confidence calibration for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. https://openaccess.thecvf.com/content_CVPRW_2020/html/w20/Kuppers_Multivariate_Confidence_Calibration_for_Object_Detection_CVPRW_2020_paper.html.

[72] Cover, T. M. and Thomas, J. A. *Entropy, Relative Entropy, and Mutual Information*, chapter 2, pages 13–55. John Wiley & Sons, Ltd, 2005. https://onlinelibrary.wiley.com/doi/abs/10.1002/047174882X.ch2, doi:https://doi.org/10.1002/047174882X.ch2.

[73] DeGroot, M. H. and Fienberg, S. E. The comparison and evaluation of forecasters. *The Statistician*, 32(1/2):12, March 1983. http://dx.doi.org/10.2307/2987588, doi:10.2307/2987588.

[74] Pakdaman Naeini, M., Cooper, G., and Hauskrecht, M. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), Feb. 2015. https://ojs.aaai.org/index.php/AAAI/article/view/9602, doi:10.1609/aaai.v29i1.9602.

[75] Lewis, D. Causal explanation. In *Philosophical Papers, Volume II*, pages 214–240. Oxford University Press, 1986. doi:10.1093/0195036468.001.0001.

[76] Sakoe, H. and Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978. doi:10.1109/TASSP.1978.1163055.

[77] Kantorovich, L. V. Mathematical methods of organizing and planning production. *Management Science*, 6(4):366–422, July 1960. http://dx.doi.org/10.1287/mnsc.6.4.366, doi:10.1287/mnsc.6.4.366.

[78] Vahdat, A. and Kautz, J. NVAE: A deep hierarchical variational autoencoder. In *Advances in Neural Information Processing Systems*, volume 33, pages 19667–19679. Curran Associates, Inc., 2020. https://proceedings.neurips.cc/paper_files/paper/2020/file/e3b21256183cf7c2c7a66be163579d37-Paper.pdf.

[79] Horvat, C. and Pfister, J.-P. Denoising normalizing flow. In *Advances in Neural Information Processing Systems*, volume 34, pages 9099–9111. Curran Associates, Inc., 2021. https://proceedings.neurips.cc/paper_files/paper/2021/file/4c07fe24771249c343e70c32289c1192-Paper.pdf.

[80] Järvinen, A. E., Kit, A., Poels, Y. R. J., Wiesen, S., Menkovski, V., Frassinetti, L., Dunne, M., ASDEX Upgrade Team, and JET Contributors. Representation learning algorithms for inferring machine independent latent features in pedestals in JET and AUG. *Physics of Plasmas*, 31(3):032508, 03 2024. doi:10.1063/5.0177005.

[81] Kit, A., Järvinen, A. E., Poels, Y. R. J., Wiesen, S., Menkovski, V., Fischer, R., Dunne, M., and ASDEX Upgrade Team. On learning latent dynamics of the AUG plasma state. *Physics of Plasmas*, 31(3):032504, 03 2024. doi:10.1063/5.0174128.

[82] Galperti, C., Felici, F., Vu, T., Sauter, O., Carpanese, F., Kong, M., Marceca, G., Merle, A., Pau, A. and Perek, A. et al. Overview of the TCV digital real-time plasma control system and its applications. *Fusion Engineering and Design*, 208:114640, 2024. https://www.sciencedirect.com/science/article/pii/S0920379624004915, doi:https://doi.org/10.1016/j.fusengdes.2024.114640.

[83] Defazio, A., Yang, X. A., Khaled, A., Mishchenko, K., Mehta, H., and Cutkosky, A. The road less scheduled. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. https://openreview.net/forum?id=0XeNkkENuI.

[84] Hendrycks, D. and Gimpel, K. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016. arXiv:1606.08415, doi:10.48550/ARXIV.1606.08415.

Appendix A. Settings and hyperparameters

Training. The model training procedure consists of minimizing the combined loss function defined in Equation 31. An optimization step consists of sampling a sequence of input timewindows $\mathbf{x}^{t_{m-w}:t_m} \in \mathbb{R}^{U\times w}$ and disruption risk labels $y^{t_m} \in \mathbb{R}$. Specifically, we gather sequences of inputs:

$$\left\{\left(\mathbf{x}^{t_{m_i-w}:t_{m_i}},\, y^{t_{m_i}}\right)\right\}_{i=1}^{N}, \quad \text{where } m_i = m_0 + is, \tag{A.1}$$

for arbitrary starting timestep m_0, from which we iterate with stride s for sequences of length N. Then, we iteratively produce outputs following the procedure described in Section 4.2. Notably, for the first timestep we do not have a previous latent position $\boldsymbol{\mu}$ available. Here, we use a separate encoder $\boldsymbol{\mu}^{t_m} = f_{\phi,0}(\mathbf{x}^{t_{m-w}:t_m})$ that is otherwise identical to f_ϕ (Equation 16), with the exception of dropping the dependence on the previous state. We do not update model parameters using the entire sequence at once, but rather iteratively detach the gradients during this procedure. The main dataset parameters are repeated in Table A.1, whereas the optimization-related settings are provided in Table A.2. The weights of the loss terms are a = 2, b = 5, c = 100 and d = 10, respectively. Details on the choice of all hyperparameters is given below in *hyperparameter choice*.

Distribution parameters. For the prior, we scan a set of structures for a single Gaussian centered at (0, 0), surrounded by a set of Gaussians spaced equidistant around this center point. We also explored optimizing the structure of the prior, but found that the procedure often led to undesirable solutions, such as several peaks merging into one or peaks completely separating from each other. Consequently, we leave the adaptive-prior setting for future work.

To quantify the effect of adding prior modes, we compute the mutual information [72] between the identified states and a set of event detections [26, 32] (Figures 11 and B.1). This metric can be interpreted as the extent to which knowledge of the state helps with estimating the odds of observing an event. We compute this metric for a range of $K \in [3, 10]$ prior modes and aggregate over all other varied parameters in our search, see Figure A.1. Here, we see that the added information starts to level off at around $K = 8$ components, and subsequently select this quantity for our model.

Remaining distribution hyperparameters are provided in Table A.3. Lastly, we provide details on the cluster assignment, where the implementation slightly deviates from the likelihood formulation. We use the inverse distance to the prior means with a scaling hyperparameter, which is equivalent for the most likely component, but reformulates the spread for the full distribution. The implementation is as follows:

$$\mathcal{C} := p(\mathcal{C} = k|\mathbf{z}) = \frac{\exp\left(\frac{1}{\tau\,(1+\|\mathbf{z}-\boldsymbol{\mu}_k\|^2)}\right)}{\sum_{j=1}^{K}\exp\left(\frac{1}{\tau\,(1+\|\mathbf{z}-\boldsymbol{\mu}_j\|^2)}\right)}, \tag{A.2}$$

which is equivalent to the Softmax of inverse distances, with 1 added to the denominator for numerical stability as the distance goes to 0, and a logistic scaling parameter τ as hyperparameter. We set τ to 0.05 for all models.

Hyperparameter choice. Optimizing the model hyperparameters is challenging due to the many conflicting objectives. Additionally, it is nontrivial to capture the quality of the identified latent space in a single metric, which limits the ability to use automated parameter optimization tools. Instead, we systematically scan a variety of parameter ranges for the disruption risk parameters ($A \in \{0.05, 0.1, 0.15\}$s, $B \in \{0.1, 0.5, 1.0\}$s) and prior modes ($K \in [3, 10]$). Additionally, we scale the neural network architecture parameters by $\{\cdot\frac{1}{4}, \cdot\frac{1}{2}, 1\}$. Loss weights a, b, c and d are tuned by hand. Models were evaluated by manually inspecting the latent space distributions, and subsequently selecting a model by three properties: (1) smooth projections of sequential measurements; (2) assigning some probability density to different peaks; and (3) clear separation w.r.t. the disruption risk variable. Unfortunately, manual parameter optimization means we are likely selecting suboptimal settings, making the development of quantitative measures for 'latent space quality' of interest for future works.

An overview of the specific model architectures and parameters is given in Table A.4 for the initial-timestep encoder, Table A.5 for the regular encoder, Table A.6 for the decoder and Table A.7 for the disruption risk network.

Parameter	Value
Stride s	50
Timewindow size w	50
1-labels before t_D, A	0.15 s
0-1 ramp before t_D, B	1.0 s

Table A.1: Data-related hyperparameters.

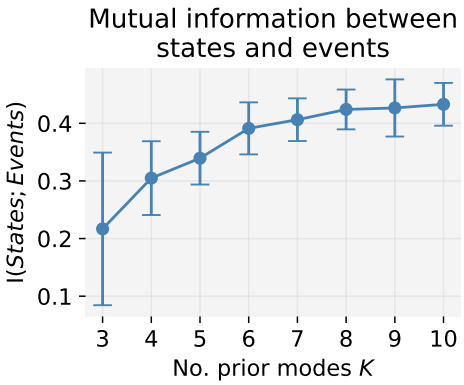

Figure A.1: The mutual information between event detections and the plasma state assignment $\mathcal{C}$, for a variety of structures for $p(\mathbf{z})$; we aggregate over other varied model parameters, indicated by the error bars (standard deviation). Each prior consists of one Gaussian centered at (0, 0), with $K - 1$ Gaussians spaced equidistant around this center point.

Parameter	Value
Optimizer	Schedule-free Adam [83]
Optimizer learning rate	0.005
Optimizer warmup steps	50
Unroll steps (N)	200
Gradient detach interval	Every 25 steps
Epochs	100
Batch size	512

Table A.2: Training-related hyperparameters.

Parameter	Value
Radius of surrounding prior components	1.5
Total number of prior components	8
Prior variance σ_{p}^2	0.1
Encoder variance σ_ϕ^2	0.03
Decoder variance σ_θ^2	0.05

Table A.3: Distribution-related hyperparameters.

Layer	Details
Input	$\mathbf{x}^{t_m-w:t_m} \in \mathbb{R}^{U\times w}$
FNO Layer	$20 \to 64$ channels, 8 modes, ReLU activation
FNO Layer	$64 \to 64$ channels, 8 modes, ReLU activation
Max pooling	Kernel size of 2
Fully connected	$1600 \to 2$
Output	$\boldsymbol{\mu}^{t_m} \in \mathbb{R}^2$

Table A.4: Architecture of the encoder for the initial timestep, $\boldsymbol{\mu}^{t_m} = f_{\phi,0}(\mathbf{x}^{t_m-w:t_m})$.

Layer	Details
Input$_1$	$\mathbf{x}^{t_{m-w}:t_m} \in \mathbb{R}^{U \times w}$
FNO Layer	$20 \to 64$ channels, 8 modes, ReLU activation
FNO Layer	$64 \to 64$ channels, 8 modes, ReLU activation
Max pooling	Kernel size of 2
Fully connected	$1600 \to 128$
Input$_2$	Concatenate $\boldsymbol{\mu}^{t_{m-s}} \in \mathbb{R}^2$
MLP$_{out}$	[128 + 2] $\to 128 \to 2$, GELU activation [84]
Output	$\Delta\boldsymbol{\mu}^{t_m} \in \mathbb{R}^2$

Table A.5: Architecture of the residual encoder $\boldsymbol{\mu}^{t_m} = \boldsymbol{\mu}^{t_{m-s}} + f_\phi(\boldsymbol{\mu}^{t_{m-s}}, \mathbf{x}^{t_{m-w}:t_m})$.

Layer	Details
Input	$\mathbf{z}^{t_m} \in \mathbb{R}^2$
Positional Encoding	$\gamma(\mathbf{z}^{t_m}) \in \mathbb{R}^8$, $L = 2$ frequencies
MLP	$8 \to 4096 \to 4096 \to 20$, ReLU activation
Output	$\widetilde{\mathbf{x}}^{t_m} \in \mathbb{R}^U$

Table A.6: Architecture of the decoder $\widetilde{\mathbf{x}}^{t_m} = f_\theta(\mathbf{z}^{t_m})$.

Layer	Details
Input	$\mathbf{z}^{t_m} \in \mathbb{R}^2$
Positional Encoding	$\gamma(\mathbf{z}^{t_m}) \in \mathbb{R}^8$, $L = 2$ frequencies
MLP	$8 \to 512 \to 1$, ReLU activation
Output activation	Sigmoid
Output	$D_{risk} \in \mathbb{R}$

Table A.7: Architecture of the disruption risk network $D_{risk} = f_D(\mathbf{z}^{t_m})$.

Appendix B. Extra results

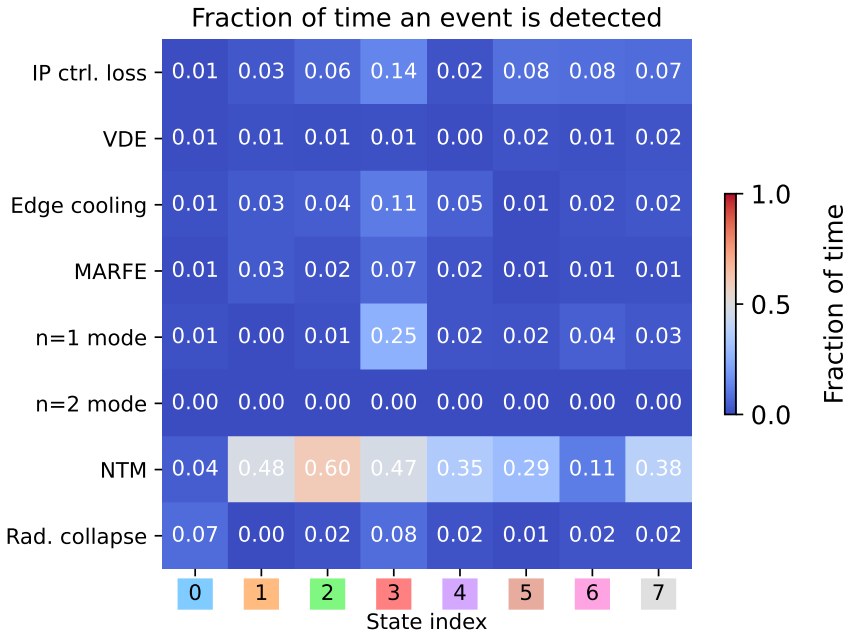

Figure B.1: Correlation between event detections computed using [26] and the states found in $\mathbf{z}$. With the exception of NTMs, most states are only detected for small fractions of the total plasma duration. Most likely, the disruption-related events operate on smaller timescales than the global plasma regimes modeled by the latent variable $\mathbf{z}$.