Contents of Appendix

A Pseudocode of SLDG

Algorithm 1: Training and Inference for SLDG.

// Training
Require: Source training data from P_{tr}
Pre-train patient encoder $E(\cdot)$ on the same task with binary cross-entropy loss for 40 epochs
for iteration ranging from 1 to 3 **do**
  Perform decoupled domain discovery with the encoder $E(\cdot)$ by Eq. (2), (3)
  Initialize gating and classifier weights $\mathbf{w}_{t,k}$, $\mathbf{w}^{+}_{t,k}$, $\mathbf{w}^{-}_{t,k}$ by Eq. (10), (8)
  for epoch ranging from 1 to 20 **do**
    for each patient $(x, y) \sim P_{tr}$ **do**
      Obtain decoupled patient representations $\{\mathbf{h}_t\}_{t\in\mathcal{T}}$ by Eq. (4), (5)
      Compute domain-specific predictions $C_{t,k}(\mathbf{h}_t)$ by Eq. (7)
      Compute gating weights $G_t(\mathbf{h}_t)$ by Eq. (9)
      Obtain final prediction o by Eq. (6)
      Update model parameters with binary cross-entropy loss
    end for
  end for
end for
// Inference
Require: Target testing data from P_{te}
for each patient $(x, y) \sim P_{tE}$ **do**
  Obtain decoupled patient representations $\{\mathbf{h}_t\}_{t\in\mathcal{T}}$ by Eq. (4), (5)
  Compute domain-specific predictions $C_{t,k}(\mathbf{h}_t)$ by Eq. (7)
  Compute gating weights $G_t(\mathbf{h}_t)$ by Eq. (9)
  Obtain final prediction o by Eq. (6)
end for

B Additional Experimental Setup

B.1 Datasets

For both datasets, we select our cohorts by filtering out visits of patients younger than 18 or older than 89 years old, visits that last longer than 10 days, and visits with data from less than 3 or more than 256 timestamps. In the case of the eICU dataset, we additionally exclude visits lasting shorter than 12 hours, as the predictions are made 12 hours after admission. Similarly, for the MIMIC-IV dataset, we exclude visits where the patient ultimately passed away, as the predictions are made upon discharge. Tab. 4 provides detailed statistics of the two datasets.

B.2 Clinical Predictive Tasks

We focus on two common clinical predictive tasks: readmission prediction and mortality prediction.

In the case of the eICU dataset, the predictions are made 12 hours after admission. Readmission prediction aims to determine whether a patient will be readmitted within the next 15 days following discharge. Mortality prediction, on the other hand, aims to predict whether a patient will pass away upon discharge. The overall prevalence for these tasks is 15% for readmission and 4% for mortality.

For the MIMIC-IV dataset, the predictions are made at the time of discharge. Similar to the eICU dataset, the readmission prediction task is defined as predicting whether a patient will be readmitted within 15 days after discharge. To prevent information leakage, the mortality prediction task for MIMIC-IV is defined as predicting whether a patient will pass away within 90 days after discharge. The overall prevalence for these tasks is 14% for readmission and 4% for mortality.

Table 4: Dataset statistics.

Item	eICU	MIMIC-IV
#Patients	116075	156549
#Admissions	149227	353238
Readmission Rate	0.15	0.14
Mortality Rate	0.04	0.04
	Region: Midwest	**Year: 2008-2010**
#Patients	29767	37328
#Admissions	35989	56433
Readmission Rate	0.10	0.14
Mortality Rate	0.03	0.04
Age	62	56
Gender	F: 0.46, M: 0.54	F: 0.53, M: 0.47
Race	African American: 0.09, Asian: 0.01, Caucasian: 0.83, Hispanic: 0.01, Native American: 0.01, Other: 0.04	African American: 0.15, Asian: 0.03, Caucasian: 0.71, Hispanic: 0.06, Native American: 0.00, Other: 0.04
Average #Events	90.01	31.87
	Region: Northeast	**Year: 2011-2013**
#Patients	5886	39125
#Admissions	6958	62586
Readmission Rate	0.17	0.15
Mortality Rate	0.06	0.04
Age	62	57
Gender	F: 0.44, M: 0.56	F: 0.53, M: 0.47
Race	African American: 0.03, Asian: 0.01, Caucasian: 0.92, Hispanic: 0.01, Native American: 0.00, Other: 0.03	African American: 0.17, Asian: 0.03, Caucasian: 0.66, Hispanic: 0.07, Native American: 0.00, Other: 0.07
Average #Events	104.54	35.19
	Region: South	**Year: 2014-2016**
#Patients	27584	41737
#Admissions	33033	64592
Readmission Rate	0.11	0.14
Mortality Rate	0.04	0.04
Age	62	57
Gender	F: 0.46, M: 0.54	F: 0.52, M: 0.48
Race	African American: 0.21, Asian: 0.01, Caucasian: 0.68, Hispanic: 0.05, Native American: 0.00, Other: 0.03	African American: 0.17, Asian: 0.04, Caucasian: 0.66, Hispanic: 0.06, Native American: 0.00, Other: 0.07
Average #Events	84.28	36.53
	Region: West	**Year: 2017-2019**
#Patients	17670	40496
#Admissions	19803	63654
Readmission Rate	0.29	0.14
Mortality Rate	0.04	0.04
Age	63	58
Gender	F: 0.45, M: 0.55	F: 0.52, M: 0.48
Race	African American: 0.05, Asian: 0.03, Caucasian: 0.77, Hispanic: 0.05, Native American: 0.02, Other: 0.08	African American: 0.17, Asian: 0.04, Caucasian: 0.65, Hispanic: 0.06, Native American: 0.00, Other: 0.07
Average #Events	85.29	36.95

B.3 Data Split

The eICU dataset comprises data collected from hospitals across the United States, while the MIMIC-IV dataset spans a period of ten years. Therefore, we utilize the eICU dataset to evaluate the model's performance across spatial gaps, and the MIMIC-IV dataset to assess its performance across temporal gaps.

For the eICU dataset, we divide it into four spatial groups based on regions: Midwest, Northeast, West, and South. Each group is then split into 70% for training, 10% for validation, and 20% for testing. We evaluate the gap between groups by comparing the performance of the backbone model trained on data from within the same group and data from outside the group. The target testing data is selected as the group (Midwest) that exhibits the largest performance gap, while the remaining groups (Northeast, West, and South) are used as the source training data.

Regarding the MIMIC-IV dataset, we divide it into four temporal groups: 2008-2010, 2011-2013, 2014-2016, and 2017-2019. Each group is further split into training, validation, and testing sets with a ratio of 70%, 10%, and 20% respectively. We consider patients admitted after 2014 as the target testing data, while all preceding patients are included in the source training data.

B.4 Baselines

We first compare SLDG to two naive baselines.

- **Oracle:** We directly train a backbone model on the training set of the target domain, select the best model on the target validation set, and evaluate its performance on the target testing set. This model is trained with in-domain data and can be seen as a upper bound for all domain generalization method.
- **Base:** We train a backbone model on the training set of the source domain, select the best model on the source validation set, and evaluate its performance on the target testing set. This model is trained with out-domain data and should act as a performance lower bound.

We then compare SLDG to both classic and recent domain generalization methods. For a fair comparisons, all the methods below are trained on the source training set, selected on the source validation set, and tested on the target testing set.

- **DANN [16]:** Domain-Adversarial Neural Networks leverage a domain classifier and a gradient reversal layer to extract domain-invariant representations. This method uses the coarse regional and temporal groups as the domain definition.
- **MLDG [22]:** Meta-Learning for Domain Generalization adopts the Model-Agnostic Meta-Learning (MAML) [15] framework and simulates the new domain scenario during training. This method also uses the coarse regional and temporal groups as the domain definition.
- **ManyDG [56]:** Many-Domain Generalization disentangles domain-variant and invariant features through mutual reconstruction and orthogonal projection. This method treats each patient as a unique domain.
- **IRM [4]:** Invariant Risk Minimization learns domain-invariant representations by minimizing a bound on the expected generalization error under domain shifts. It acts as a regularizer and does not require domain IDs.
- **MMLD [29]:** Domain Generalization using a Mixture of Multiple Latent Domains iteratively assigns pseudo domain labels via clustering and trains the domain-invariant feature extractor through adversarial learning. This method does not rely on domain IDs.
- **DRA [14]:** Latent Domain Learning with Dynamic Residual Adapters uses layer-wise multi-head correction networks with a gating mechanism and residual connection to enhance model learning. This method does not rely on domain IDs.

B.5 Implementation Details

For all baselines, we use the Transformer as the backbone encoder. The number of layers is 3, the embedding dimension is 128, the number of attention heads is 2. The event embedding look-up table is initialized with ClinicalBERT [3] embeddings of the event name and then project it down to 128 dimension with a linear layer. Patient demographics features (age, gender, and ethnicity) are separately embeded with another embedding look-up table. We also embed the timestamps with sinusoidal positional encoding. The medical, patient demographics, and temporal embeddings are added together to form the overall sequence embedding. For SLDG, UMAP [30] from UMAP-learn [41] is used with 2 components, 10 neighbors, and 0 minimum distance; and k-Means from Scikit-learn [35] is used with the default hyper-parameter. We apply a dropout of rate 0.2. We use Adam as the optimizer with a learning rate of 1e-4 and a weight decay of 1e-5. All models are trained for 100 epochs. The batch size is 256. We select the best model by monitoring the AUPRC score on the source validation set (except for the Oracle baseline, where we directly use the target validation set). We implement SLDG using PyTorch [34] 1.11 and Python 3.8. The model is trained on a CentOS Linux 7 machine with 128 AMD EPYC 7513 32-Core Processors, 512 GB memory, and eight NVIDIA RTX A6000 GPUs.

C Limitations and Broader Impacts

In terms of limitations, it is important to acknowledge that our work operates under the assumption that the target testing data still exhibit some similarities with the source training data. If there is

a significant distribution shift, the knowledge acquired from the source training data may become irrelevant. In such cases, neither the DG baselines nor our proposed method can effectively address the problem. It would be more appropriate to explore transfer learning or train a new model to obtain a better solution. Further, we propose SLDG to tackle two main challenges: (1) unknown domain IDs and (2) distinct characteristics across domains. In the scenario when the domain IDs are given and clearly separable (e.g., photo, art painting, cartoon, and sketch in the PACS [21] dataset), SLDG 's domain discovery approach might be unnecessary. Existing DG methods directly utilizing the domain IDs could be a better solution.

In terms of broader impacts, our work tackles a practical and prevalent issue in healthcare known as the domain shift problem. We aim to inspire future research in this area: both by investigating the existence of domain shift under various scenarios, and by contributing to the development of effective solutions for this real-world challenge.

D Notations

Notation	Meaning
x	a patient's hospital visit
$[e_1, \ldots, e_m]$	sequence of m events
t	type of an event
$T(\cdot)$	mapping function from event to its type
$\mathcal{E}$	set of all events
$\mathcal{T}$	set of all event types
$y \in \{+, -\}$	label, i.e., the occurrence of a certain future event
$f_\phi(\cdot)$	overall clinical predictive model
ϕ	model parameter
$l(\cdot)$	loss function
P_{tr}, P_{te}	training and testing data distribution
$E_t(\cdot)$	feature-specific patient encoder for event type t
$\mathbf{h}_t$	patient representation in latent space of type t
h	hidden dimension
$\{\mathbf{h}_t^{(i)}\}_{i=1}^{N_{tr}}$	all patient representations in latent space of type t
N_{tr}	total number of training samples
K_t	number of discovered domains in the latent space of type t
$\mathbf{M}_t$	domain assignment matrix
$[\mathbf{e}_1, \ldots, \mathbf{e}_m]$	contextualized representation for event sequence $[e_1, \ldots, e_m]$
$E(\cdot)$	embedding function
$\{\mathbf{h}_t\}_{t \in \mathcal{T}}$	multi-vector representations for a single patient
$C_{t,k}(\cdot)$	customized classifier for the discovered domain k in the latent space of type t
$G_{t,k}(\cdot)$	the gating weight for the customized classifier $C_{t,k}(\cdot)$
o	model output
$\mathbf{w}_{t,k}^+, \mathbf{w}_{t,k}^-$	learnable prototype weight vectors of the positive and negative classes for the k-th discovered domain in the latent space of type t
$d(\cdot, \cdot)$	Euclidean distance
$\mathbf{w}_{t,k}$	learnable prototypical weight vector for the discovered domain k in the latent space of type t

E Additional Illustrations

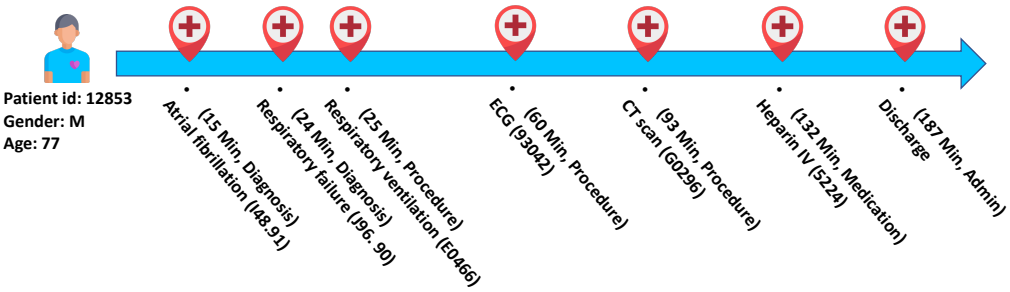

Figure 6: An illustration of the patient visit as input.

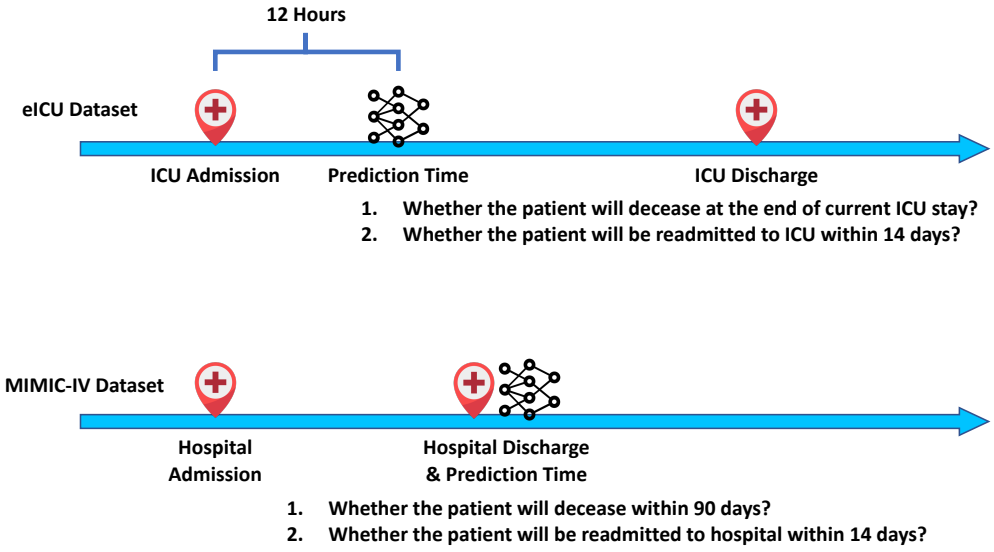

Figure 7: An illustration of the task definitions in the eICU and the MIMIC-IV datasets.

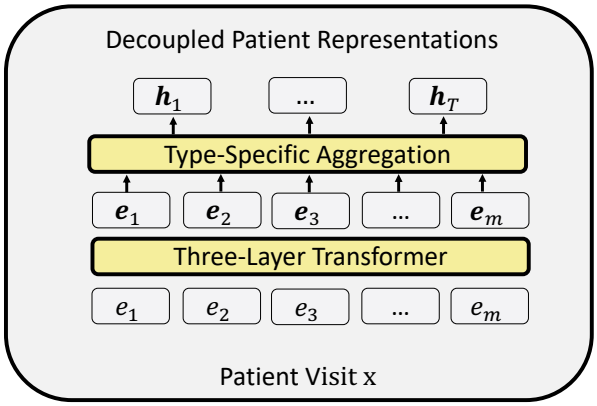

Figure 8: The architecture of the feature-specific patient encoder.