A Simplified Generative Counterfactual Framework for Single-Treatment Settings

[Author Name]

1 Problem Setup and Notation

Consider n independent observational units indexed by $i = 1, \ldots, n$. Each unit corresponds to a *patient–time-window* or *patient–visit* pair. For unit i we observe:

- a feature vector $X_i \in \mathbb{R}^p$, which may include baseline covariates and time-varying clinical information;
- a *single* treatment assignment T_i, where $T_i \in \{0, 1\}$ for binary treatment or $T_i \in \{1, \ldots, K\}$ for multi-category treatment;
- an outcome $Y_i \in \mathbb{R}$, for example a change in a clinical score between visits.

The *single-treatment assumption* states that for each unit i and each time window there is exactly one treatment value T_i. A patient may switch treatments across different visits, but in one time window the treatment is unique.

There exists an unobserved confounder U_i that affects both treatment assignment and the outcome. We consider the following structural causal model (SCM):

$$\begin{aligned} U_i &\sim P_U, \\ X_i &= g_X(U_i, \varepsilon_{X,i}), \\ T_i &= g_T(X_i, U_i, \varepsilon_{T,i}), \\ Y_i &= g_Y(T_i, X_i, U_i, \varepsilon_{Y,i}), \end{aligned} \tag{1}$$

where $\varepsilon_{X,i}, \varepsilon_{T,i}, \varepsilon_{Y,i}$ are mutually independent exogenous noise variables.

For each t in the treatment space $\mathcal{T}$, let $Y_i(t)$ denote the potential outcome that would be observed for unit i if the treatment were set to t. We assume standard SUTVA (stable unit treatment value assumption) and well-defined potential outcomes.

For a given covariate vector x, the conditional mean potential outcome is

$$\mu(t, x) = \mathbb{E}\big[Y_i(t) \mid X_i = x\big], \tag{2}$$

and the individual treatment effect (ITE) at x between t and t' is

$$\tau(t', t \mid x) = \mathbb{E}\big[Y_i(t') - Y_i(t) \mid X_i = x\big]. \tag{3}$$

The average treatment effect (ATE) between two treatment levels t', t is

$$\mathrm{ATE}(t', t) = \mathbb{E}\big[Y_i(t') - Y_i(t)\big]. \tag{4}$$

Our goal is to estimate $\tau(t', t \mid x)$ and $\mathrm{ATE}(t', t)$ from observational data $\{(X_i, T_i, Y_i)\}_{i=1}^n$ under unobserved confounding U_i.

2 Representation Assumptions

We introduce a representation map

$$\Phi_\varphi : \mathbb{R}^p \to \mathbb{R}^{d_s} \times \mathbb{R}^{d_c}, \quad \Phi_\varphi(X) = (S, C),$$

with parameters φ. The vector S represents *stable clinical state* and C represents *selection and confounding signal*.

2.1 Factorization Assumption

We aim to construct Φ_φ such that the following approximate factorization holds:

$$\begin{aligned} S_i &\approx s(X_i), \\ C_i &\approx h(X_i, U_i), \end{aligned} \tag{5}$$

for some functions s and h. We interpret S_i as absorbing stable, relatively invariant aspects of the patient state, and C_i as a proxy for the unobserved confounder U_i that influences treatment assignment.

2.2 Approximate Sufficiency for Outcome

We assume that the pair (S_i, C_i) is approximately sufficient for the outcome in the following sense:

$$Y_i(t) \perp\!\!\!\perp X_i \mid S_i, C_i \quad \text{for all } t \in \mathcal{T}, \tag{6}$$

up to approximation error due to representation learning. Condition (6) means that, given (S_i, C_i) and the treatment level, there is no remaining predictive information about $Y_i(t)$ in the raw covariates X_i.

2.3 Approximate Propensity Representation

Let

$$e(x) = \mathbb{P}(T = 1 \mid X = x) \tag{7}$$

denote the propensity score in the binary case. We introduce a propensity head

$$\pi_\beta : \mathbb{R}^{d_c} \to [0, 1],$$

parameterized by β, and we aim to approximate

$$\pi_\beta(C_i) \approx e(X_i). \tag{8}$$

Thus, C_i is constructed to carry the information that is relevant for treatment assignment.

3 Factorized Causal Representation Learning

We describe a self-supervised and semi-supervised objective that encourages the desired structure of (S_i, C_i).

3.1 Data Augmentation and Encoders

For each observation X_i, we apply two weak data augmentations

$$X_i^{(1)} = a_1(X_i), \qquad X_i^{(2)} = a_2(X_i), \tag{9}$$

where a_1, a_2 are random transformations that preserve clinical semantics (for example small additive noise or small scaling). We encode

$$(S_i^{(1)}, C_i^{(1)}) = \Phi_\varphi(X_i^{(1)}), \qquad (S_i^{(2)}, C_i^{(2)}) = \Phi_\varphi(X_i^{(2)}). \tag{10}$$

For the propensity head we compute

$$P_i^{(1)} = \pi_\beta(C_i^{(1)}), \qquad P_i^{(2)} = \pi_\beta(C_i^{(2)}). \tag{11}$$

3.2 Stability Loss for S

We enforce invariance of the stable representation under small augmentations:

$$\mathcal{L}_S = \frac{1}{n} \sum_{i=1}^{n} \left\| S_i^{(1)} - S_i^{(2)} \right\|_2^2. \tag{12}$$

3.3 Propensity Consistency Loss for C

We want the propensity prediction to be consistent across the two augmented views. This encourages $C_i^{(1)}$ and $C_i^{(2)}$ to retain treatment-relevant information. We define

$$\mathcal{L}_{\text{prop-cons}} = \frac{1}{n} \sum_{i=1}^{n} \left(P_i^{(1)} - P_i^{(2)} \right)^2. \tag{13}$$

3.4 Propensity Fitting Loss

Using the observed treatment labels T_i, we fit the propensity head on the full features:

$$(S_i, C_i) = \Phi_\varphi(X_i), \tag{14}$$

and

$$P_i = \pi_\beta(C_i). \tag{15}$$

For binary treatment we use the cross-entropy loss:

$$\mathcal{L}_{\text{prop-fit}} = -\frac{1}{n} \sum_{i=1}^{n} \Big[T_i \log P_i + (1 - T_i) \log(1 - P_i) \Big]. \tag{16}$$

For multi-class treatment we can replace π_β by a softmax head and (16) by the standard multi-class cross-entropy.

3.5 Decorrelation Between S and C

To encourage factorization we penalize linear correlation between S and C. Consider a mini-batch of size B with centered matrices

$$\tilde{S} \in \mathbb{R}^{B\times d_s}, \quad \tilde{C} \in \mathbb{R}^{B\times d_c},$$

obtained by subtracting the batch mean from each representation. We define the cross-covariance matrix

$$\mathrm{Cov}(S, C) = \frac{1}{B}\tilde{S}^\top \tilde{C} \in \mathbb{R}^{d_s\times d_c}. \tag{17}$$

The decorrelation loss is

$$\mathcal{L}_{\text{decouple}} = \big\|\mathrm{Cov}(S, C)\big\|_F^2 = \sum_{a=1}^{d_s}\sum_{b=1}^{d_c}\big(\mathrm{Cov}(S_a, C_b)\big)^2. \tag{18}$$

3.6 Variance Regularization

To avoid collapse of the representations, we constrain the batch-wise standard deviation of each dimension to stay above a threshold $v > 0$. Let $\mathrm{std}(S_{\cdot k})$ denote the standard deviation of the k-th coordinate across the batch. We define

$$\mathcal{L}_{\text{var}} = \sum_{k=1}^{d_s}\max\big\{0, v - \mathrm{std}(S_{\cdot k})\big\}^2 + \sum_{k=1}^{d_c}\max\big\{0, v - \mathrm{std}(C_{\cdot k})\big\}^2. \tag{19}$$

3.7 Total Representation Loss

The total representation loss is

$$\mathcal{L}_{\text{SSL}}(\varphi, \beta) = \lambda_S\mathcal{L}_S + \lambda_{\text{pc}}\mathcal{L}_{\text{prop-cons}} + \lambda_{\text{pf}}\mathcal{L}_{\text{prop-fit}} + \lambda_{\text{dec}}\mathcal{L}_{\text{decouple}} + \lambda_{\text{var}}\mathcal{L}_{\text{var}}, \tag{20}$$

where $\lambda_S, \lambda_{\text{pc}}, \lambda_{\text{pf}}, \lambda_{\text{dec}}, \lambda_{\text{var}} > 0$ are tuning weights.

We minimize $\mathcal{L}_{\text{SSL}}(\varphi, \beta)$ over (φ, β) to obtain the encoder Φ_φ and the propensity head π_β. After training, we fix Φ_φ and use

$$(S_i, C_i) = \Phi_\varphi(X_i) \quad \text{for all } i.$$

4 Outcome Model on the Factorized Representation

We model the outcome as a function of the factorized representation and the treatment:

$$Y_i = f_\theta(S_i, C_i, T_i) + \varepsilon_i, \tag{21}$$

where $f_\theta : \mathbb{R}^{d_s} \times \mathbb{R}^{d_c} \times \mathcal{T} \to \mathbb{R}$ is a neural network with parameters θ, and ε_i is mean-zero noise.

A convenient parameterization in the binary case is

$$f_\theta(S, C, T) = m_\theta(S, C) + \tau_\theta(S, C)\cdot T, \tag{22}$$

where m_θ and τ_θ are neural networks. Then the individual treatment effect at representation (S, C) is directly given by $\tau_\theta(S, C)$.

We estimate θ by minimizing the empirical risk

$$\mathcal{L}_Y(\theta) = \frac{1}{n}\sum_{i=1}^{n}\big(Y_i - f_\theta(S_i, C_i, T_i)\big)^2. \tag{23}$$

5 Confounding Diffusion on the Selection Representation

We now describe a conditional diffusion model on the confounding representation C_i. The diffusion model learns the conditional distribution $p(C \mid S, T)$. We use this model to generate counterfactual confounding representations under alternative treatments.

5.1 Forward Diffusion Process

For each unit i, we set the initial state

$$c_0 = C_i \in \mathbb{R}^{d_c}. \tag{24}$$

We choose a noise schedule $\{\beta_t\}_{t=1}^{T}$ with $0 < \beta_t < 1$ and define

$$\alpha_t = 1 - \beta_t, \qquad \bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s. \tag{25}$$

The forward diffusion process is

$$q(c_t \mid c_{t-1}) = \mathcal{N}\big(\sqrt{\alpha_t}\, c_{t-1},\, \beta_t I\big), \quad t = 1, \ldots, T. \tag{26}$$

This implies the closed form

$$q(c_t \mid c_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\, c_0,\, (1 - \bar{\alpha}_t) I\big). \tag{27}$$

5.2 Noise-Prediction Network and Training Objective

We define a noise-prediction network

$$\varepsilon_\psi : \mathbb{R}^{d_c} \times \{1, \ldots, T\} \times \mathbb{R}^{d_s} \times \mathcal{T} \to \mathbb{R}^{d_c}, \quad (c_t, t, S, T) \mapsto \varepsilon_\psi(c_t, t, S, T),$$

with parameters ψ.

For training, we sample

- a data index i uniformly from $\{1, \ldots, n\}$;
- a time index t uniformly from $\{1, \ldots, T\}$;
- a noise vector $\varepsilon \sim \mathcal{N}(0, I)$.

We construct

$$c_t = \sqrt{\bar{\alpha}_t} C_i + \sqrt{1 - \bar{\alpha}_t}\, \varepsilon. \tag{28}$$

The diffusion loss is the mean squared error between the true noise and the predicted noise:

$$\mathcal{L}_{\text{diff}}(\psi) = \mathbb{E}_{i,t,\varepsilon} \big\| \varepsilon - \varepsilon_\psi(c_t, t, S_i, T_i) \big\|_2^2. \tag{29}$$

We minimize $\mathcal{L}_{\text{diff}}(\psi)$ over ψ.

5.3 Reverse Diffusion and Counterfactual Sampling

After training ε_ψ, we can sample from the conditional distribution $p(C \mid S, T = t')$ for an alternative treatment t'.

For each unit i and desired treatment $t' \in \mathcal{T}$ we perform the following reverse diffusion procedure:

To capture uncertainty, we may repeat Algorithm 1 independently S times and obtain samples

$$C_i^{\text{cf},(s)}(t'), \quad s = 1, \ldots, S. \tag{30}$$

Algorithm 1 Sampling counterfactual confounding representation $C_i^{\mathrm{cf}}(t')$

Input: S_i, target treatment t', trained ε_ψ, noise schedule $\{\alpha_t, \bar{\alpha}_t\}_{t=1}^T$.
Sample $c_T \sim \mathcal{N}(0, I)$.
for $t = T, T-1, \ldots, 1$ **do**
  Compute

$$\hat{\varepsilon} = \varepsilon_\psi(c_t, t, S_i, t').$$

  Compute the mean

$$\mu_t(c_t, S_i, t') = \frac{1}{\sqrt{\alpha_t}}\left(c_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\hat{\varepsilon}\right).$$

  if $t > 1$ **then**
    Sample $z_t \sim \mathcal{N}(0, I)$ and set

$$c_{t-1} = \mu_t(c_t, S_i, t') + \sigma_t z_t,$$

    where σ_t^2 is a predefined variance schedule.
  else
    Set $c_0 = \mu_t(c_t, S_i, t')$.
  end if
end for
Output: $C_i^{\mathrm{cf}}(t') = c_0$.

6 Counterfactual Prediction and Treatment Effect Estimation

6.1 Counterfactual Outcomes

For a given unit i and target treatment t' we define the factual prediction

$$\hat{\mu}_i(T_i) = f_\theta(S_i, C_i, T_i), \tag{31}$$

and the counterfactual prediction.

If we sample a single counterfactual confounding representation $C_i^{\mathrm{cf}}(t')$, we set

$$\hat{\mu}_i(t') = f_\theta\big(S_i, C_i^{\mathrm{cf}}(t'), t'\big). \tag{32}$$

If we sample S draws $C_i^{\mathrm{cf},(s)}(t')$, we average them:

$$\hat{\mu}_i(t') = \frac{1}{S}\sum_{s=1}^{S} f_\theta\big(S_i, C_i^{\mathrm{cf},(s)}(t'), t'\big). \tag{33}$$

6.2 Individual and Average Treatment Effects

The estimated individual treatment effect for unit i between the observed treatment T_i and an alternative treatment t' is

$$\hat{\tau}_i(t', T_i) = \hat{\mu}_i(t') - \hat{\mu}_i(T_i). \tag{34}$$

For a fixed baseline treatment t and a target treatment t', the estimated average treatment effect is

$$\widehat{\mathrm{ATE}}(t', t) = \frac{1}{n}\sum_{i=1}^{n}\Big(\hat{\mu}_i(t') - \hat{\mu}_i(t)\Big). \tag{35}$$

When $t = 0$ is a control or placebo, $\widehat{\mathrm{ATE}}(t', 0)$ measures the average effect of switching from control to treatment t'.

7 Summary of the Simplified Algorithm

The proposed framework for the single-treatment setting proceeds in three main stages:

1. **Factorized representation learning.** Train the encoder Φ_φ and the propensity head π_β by minimizing $\mathcal{L}_{\mathrm{SSL}}(\varphi, \beta)$ in (20). This yields the stable representation S_i and the selection representation C_i.

2. **Outcome model fitting.** Fix Φ_φ and estimate f_θ by minimizing $\mathcal{L}_Y(\theta)$ in (23), using inputs (S_i, C_i, T_i) and outputs Y_i.

3. **Confounding diffusion and counterfactual estimation.** Train the diffusion model on C_i via $\mathcal{L}_{\mathrm{diff}}(\psi)$ in (29). For each unit i and desired treatment t', sample counterfactual confounding representations $C_i^{\mathrm{cf}}(t')$ with Algorithm 1, evaluate counterfactual outcomes via (33), and compute individual and average treatment effects.

This design preserves the self-supervised separation of signals, keeps the diffusion-based counterfactual generation, and combines both components in a motivated way for the single-treatment-per-window causal inference problem.