Given a latent variable, $z_t \in \mathbb{R}^D$ sampled from an encoder network, $q_\phi(\cdot \mid \mathbf{x}_t)$, and a disentanglement variable, $v_t \in \mathbb{R}^N$, we seek to minimize

$L_\text{scrub}(\phi) = \max_\psi[\mathbb{E}_{\mathbf{x}_t, \mathbf{v}_t} [\mathbb{E}_{\mathbf{z_t}\sim q_\phi(\cdot\mid\mathbf{x}_t)}[\log p(\mathbf{v}_t \mid f_\psi(\mathbf{z}_t))]]]$

where $f_\psi(\cdot)$ is an adversarial decoder that aims to maximize the log-likelihood of $\mathbf{v}_t$ given $\mathbf{z}_t$. Below, we describe the process.

Algorithm 1: SC-VAE-MALS

We consider a linear decoder, $f_{\psi}(\mathbf{z}) = \psi \mathbf{z} = \mathbf{\hat{v}}$, where $\psi = \psi^{(0)}(\psi^{(1)})^{-1}$, which can be evaluated using mean squared error, $L(\mathbf{z}, \mathbf{v}; \psi) = ||\mathbf{v} - f_{\psi_a}(\mathbf{z})||^2_2$.

$\phi, \theta \leftarrow$ Initialize parameters of the network

$\psi_a, \psi_b \in \mathbb{R}^{N\times D} \leftarrow$ Initialize parameters of two linear decoders

Initialize forgetting factors with fixed offset, ϵ

$\lambda_a \leftarrow \alpha \in (0, 1-\epsilon)$

$\lambda_b \leftarrow \lambda_a + \epsilon$

repeat

Draw minibatch with K samples: $(\mathbf{x_k}, \mathbf{v_k} \in \mathbb{R}^{N\times K})$

$\mathbf{z_k} \sim q_\phi(\cdot \mid \mathbf{x_k}) \in \mathbb{R}^{D\times K}$

Calculate mean squared error for each decoder and average for scrubbing loss

$L_\text{scrub} = -\frac{1}{2}[L(\mathbf{z_k}, \mathbf{v_k}; \psi_a) + L(\mathbf{z_k}, \mathbf{v_k}; \psi_b)]$

Forgetting factors step by Δ in the direction of the better decoder between f_{ψ_a} and f_{ψ_b}

if $L(\mathbf{z_k}, \mathbf{v_k}; \psi_a) > L(\mathbf{z_k}, \mathbf{v_k}; \psi_b)$

$\lambda_a = max(\lambda_a - \Delta, 0), \lambda_b = \lambda_a + \epsilon$

else

$\lambda_b = min(\lambda_b + \Delta, 1), \lambda_a = \lambda_b - \epsilon$

end if

Update ψ_a and ψ_b based on the normal equations for ordinary least squares regression

$\psi_a = [\mathbf{v_k} \mathbf{z_k}^\top + \lambda_a \psi_a^{(0)}] \left ([\mathbf{z_k} \mathbf{z_k}^\top + \lambda_a \psi_a^{(1)}] \right)^{-1}$

$\psi_b = [\mathbf{v_k} \mathbf{z_k}^\top + \lambda_b \psi_b^{(0)}] \left ([\mathbf{z_k} \mathbf{z_k}^\top + \lambda_b \psi_b^{(1)}] \right)^{-1}$

Update network parameters

$\phi \leftarrow \phi + \nabla[L_\text{scrub} + L_\text{ELBO} + L_\text{Recon}]$

$\theta \leftarrow \theta + \nabla[L_\text{Recon}]$

until convergence

Algorithm 2: SC-VAE-QD

We consider the class-conditional Bayesian classifier, $f_{\psi}(\mathbf{z}) = p(v=c|\mathbf{z})$, with likelihood, $p(z|v=c) = Normal(\mathbf{z} | \mu^{(c)}, \Sigma^{(c)})$. For multi-class problems, we maintain the *one vs rest* estimator per class where $\psi = { \mu^{(c)}, \Sigma^{(c)}, \mu^{(c')}, \Sigma^{(c')} \ \forall \ c \in C}$. This estimator can be evaluated per class based on the Gaussian log-likelihood, $L(\mathbf{z}, v; \psi^{(c)}) = \ell (\mu_{a}^{(c)}, \Sigma_{a}^{(c)} | \mathbf{z}, v=c) - \ell (\mu_{a}^{(c')}, \Sigma_{a}^{(c')} | \mathbf{z}, v\neq c)$.

$\phi, \theta \leftarrow$ Initialize parameters of the network

$\psi_a, \psi_b \leftarrow$ Initialize parameters of two quadratic discriminants

$\lambda_a \leftarrow \alpha \overrightarrow{\mathbf{1}}_C, \ \alpha \in (0, 1-\epsilon)$

$\lambda_b \leftarrow \lambda_a + \epsilon\overrightarrow{\mathbf{1}}_C$

repeat

Draw minibatch with K samples: $(\mathbf{x_k}, \mathbf{v_k})$

$\mathbf{z_k} \sim q_\phi(\cdot \mid \mathbf{x_k}) \in \mathbb{R}^{D\times K}$

$L_\text{scrub} \leftarrow 0$

for $c \in C$

Evaluate the Gaussian log-likelihood ratio for each quadratic classifier and average for the scrubbing loss

$L_\text{scrub} = L_\text{scrub} + \frac{1}{2K} \sum_K [L(\mathbf{z}_k, v_k; \psi^{(c)}_a) + L(\mathbf{z}_k, v_k; \psi^{(c)}_b)]$

Forgetting factors step by Δ in the direction of the better classifier between f_{ψ_a} and f_{ψ_b}

if $\frac{1}{K} \sum_K L(\mathbf{z}_k, v_k; \psi^{(c)}_a) > \frac{1}{K} \sum_K L(\mathbf{z}_k, v_k; \psi^{(c)}_b)$

$\lambda_{a}^{(c)} = max(\lambda_{a}^{(c)} - \Delta, 0), \ \lambda_{b}^{(c)} = \lambda_{a}^{(c)} + \epsilon$

else

$\lambda_{b}^{(c)} = min(\lambda_{b}^{(c)} + \Delta, 1), \ \lambda_{a}^{(c)} = \lambda_{b}^{(c)} - \epsilon$

end if

Update class means and covariances for both estimators

for $i \in [a, b]$

$\mu_i^{(c)} = \mathbb{E}_{\mathbf{v_k} = c}[\mathbf{z_k}] + \lambda_{i}^{(c)} \mu^{(c)}$

$\Sigma_i^{(c)} = \text{Cov}_{\mathbf{v_k} = c}[\mathbf{z_k}, \mathbf{z_k}]$

$\mu_i^{(c')} = \mathbb{E}_{\mathbf{v_k} \neq c}[\mathbf{z_k}] + \lambda_{i}^{(c')} \mu^{(c')}$

$\Sigma_i^{(c')} = \text{Cov}_{\mathbf{v_k} \neq c}[\mathbf{z_k}, \mathbf{z_k}]$

end for

end for

Update network parameters:

$\phi \leftarrow \phi + \nabla[L_\text{scrub} + L_\text{ELBO} + L_\text{Recon}]$

$\theta \leftarrow \theta + \nabla[L_\text{Recon}]$

until convergence