Supplement for *"Semi-Generative Modelling: Covariate-Shift Adaptation with Cause and Effect Features"* (AISTATS 2019)

In this supplement we provide some additional results, plots, and derivations. Appendix A contains a derivation of how to make predictions in a linear Gaussian regression setting, given the model from the main paper and a parameter estimate θ. Appendix B illustrates on synthetic regression data how giving too much weight to the unsupervised model, $P(X_E|X_C)$, can lead to overfitting in such a setting. Appendix C contains plots for tuning the hyperparameter λ. Appendix D provides details for using our semi-generative modelling approach in a Bayesian framework. It also contains additional experiments investigating the behaviour under model-misspecification, as well as some plots of the resulting posterior distributions and decision boundaries found by the different estimators. Appendix E provides pseudo-code implementations of our approach. Finally, we also provide code which can be used to reproduce our results as a separate supplementary material.

Appendix A. Proofs

Proposition 1 *Given the linear Gaussian regression model from the main paper, and a parameter estimate $\theta = (a, b, c, d, \sigma_Y, \sigma_E)$, the most likely outcome for a new observation (x_C, x_E) is given by*

$$\hat{y} = \frac{\sigma_E^2(a + bx_C) + d^2\sigma_Y^2(\frac{x_E - c}{d})}{\sigma_E^2 + d^2\sigma_Y^2}.$$

Proof *Denoting the pdf of a normally distributed random variable with mean μ and standard deviation σ by $\phi(x|\mu, \sigma^2)$ it follows that:*

$$\begin{aligned}
y^* &= \arg\max_y \; P(y \,|\, x_C^*, x_E^*, \theta) \\
&= \arg\max_y \; \frac{P(y \,|\, x_C^*, \theta)\, P(x_E^*|y, \theta)}{P(x_E^*|x_C^*, \theta)} \\
&= \arg\max_y \; P(y \,|\, x_C^*, \theta_Y)\, P(x_E^*|y, \theta_E) \\
&= \arg\max_y \; \phi(y \,|\, a + bx_C^*, \sigma_Y^2)\, \phi(x_E^* \,|\, c + dy, \sigma_E^2) \\
&= \arg\max_y \; \phi(y \,|\, a + bx_C^*, \sigma_Y^2)\, \phi\Big(y \,|\, \frac{x_E^* - c}{d}, \frac{\sigma_E^2}{d^2}\Big) \\
&= \arg\max_y \; \phi\Big(y \,|\, \frac{\sigma_E^2(a + bx_C^*) + d^2\sigma_Y^2(\frac{x_E^* - c}{d})}{\sigma_E^2 + d^2\sigma_Y^2}, \frac{\sigma_E^2\sigma_Y^2}{\sigma_E^2 + d^2\sigma_Y^2}\Big) \\
&= \frac{\sigma_E^2(a + bx_C^*) + d^2\sigma_Y^2(\frac{x_E^* - c}{d})}{\sigma_E^2 + d^2\sigma_Y^2}
\end{aligned}$$

where the penultimate equality follows from a result about the product of two normal pdfs:

$$\phi(x \,|\, \mu_1, \sigma_1^2)\, \phi(x \,|\, \mu_2, \sigma_2^2) = \phi\Big(x \,|\, \frac{\sigma_2^2\mu_1 + \sigma_1^2\mu_2}{\sigma_1^2 + \sigma_2^2}, \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\Big)$$

■

Appendix B. Demonstration of Overfitting on Synthetic Regression Data

Figure 1 shows some additional experiments on synthetic data for illustration purposes. The overfitting problem for too flexible models is apparent from the bottom row, which shows that our estimator θ_P perfectly fits the unsupervised model $X_E \mid X_C$, but at the cost of completely mismatching the two mechanisms $Y \mid X_C$ and $X_E \mid Y$. Recalling our regression models, $Y = a + bX_C$, $X_E = c + dY$, and so $X_E = bdX_C + \text{const.}$, it is clear that the positive slope of the unsupervised model, $bd > 0$, can be explained by either $b, d < 0$, which is the true model, or by $b, d > 0$, which is the model found by θ_P. With this flexibility, it thus only seems logical that overfitting of $X_E \mid X_C$ occurs eventually when equal weight is given to labelled and unlabelled observations, i.e., using $\lambda = \frac{n_S}{n_S + n_T}$.

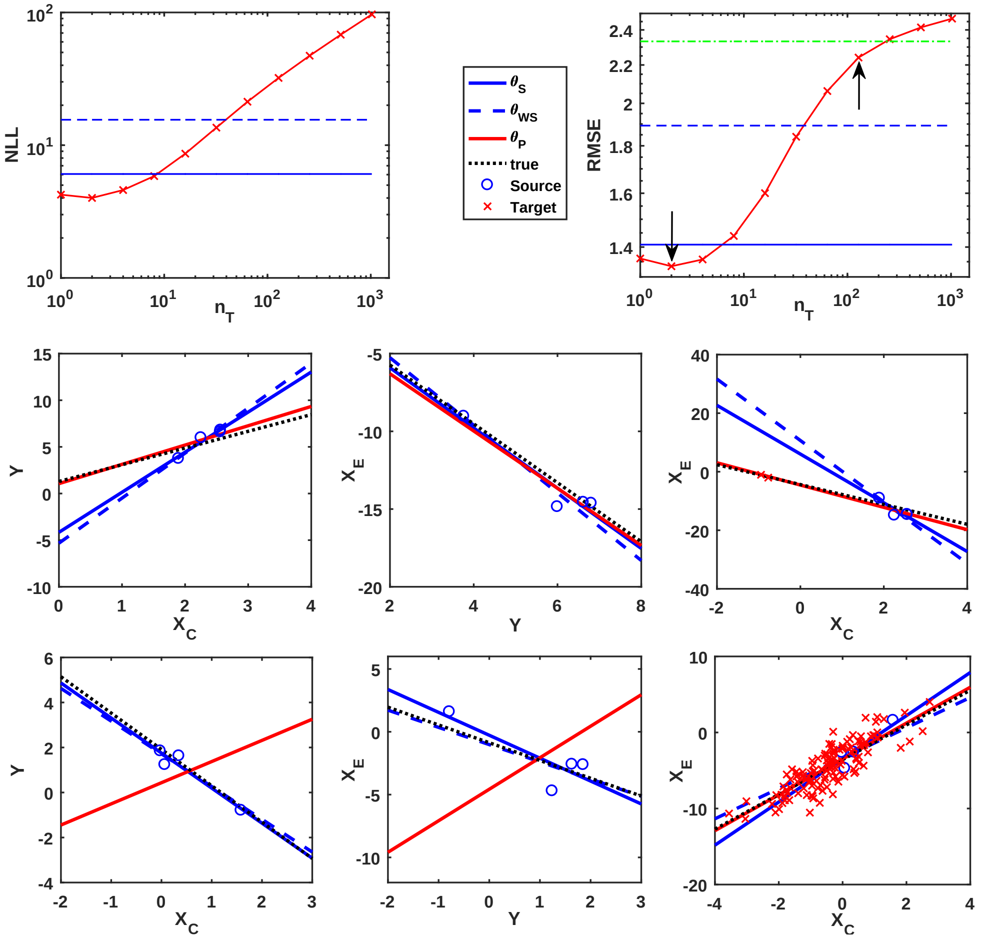
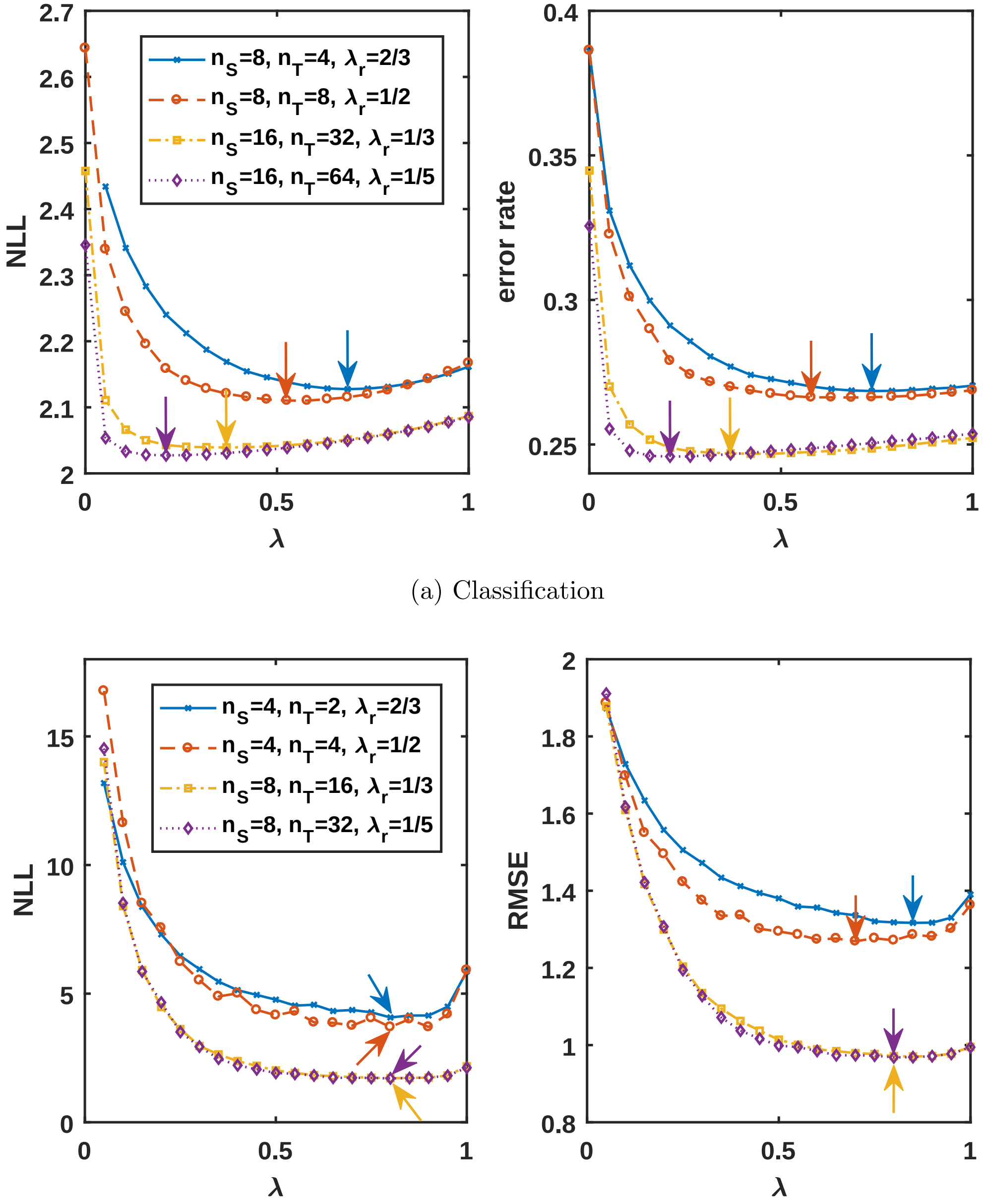


Figure 1: Synthetic regression data results for $n_S = 4$ and $\lambda = \frac{n_S}{n_S + n_T}$. Learning curves of NLL and RMSE vs n_T (top) are test set averages over 10^3 different datasets resulting from random choices of a, b, c, d, σ_Y, and σ_E. Arrows mark $n_T = 2$ with $\lambda = \frac{2}{3}$, and $n_T = 128$ with $\lambda \approx 0.03$ for which example model fits are shown in the middle and bottom rows.

Appendix C. Tuning λ

The results of our experiments for tuning the hyperparameter $\lambda \in (0,1)$ are shown in Fig. 2.



(a) Classification

(b) Regression

Figure 2: Tuning the hyperparameter λ - Shown are negative log-likelihood and RMSE/error rate against $\lambda \in (0,1)$ for different combinations of n_S and n_T (see legends); arrows mark the minima of each curve. All results are test set averages over 10^4 runs.

Appendix D. Bayesian Approach and Experiments

For a Bayesian approach, we place a rather flat (i.e., with large σ) normal prior π on θ, so as to not include much prior knowledge on how the data is generated. We can then compute the log-posterior distribution up to additive constants:

$$\log P(\theta_P \,|\, S_S, S_T) = \log \pi(\theta) + (n_S + n_T)\ell_P(\theta) + \text{const.}, \tag{1}$$

In order to make predictions for new data $(x_C^{\text{new}}, x_E^{\text{new}})$, we estimate the required integral using a Monte Carlo approximation:

$$\begin{aligned} P(Y = y \,|\, x_C^{\text{new}}, x_E^{\text{new}}) &= \int_\theta P(Y = y \,|\, x_C^{\text{new}}, x_E^{\text{new}}, \theta) P(\theta \,|\, S_S, S_T)\, \mathrm{d}\theta \\ &\approx \frac{1}{K} \sum_{k=1}^{K} P(Y = y \,|\, x_C^{\text{new}}, x_E^{\text{new}}, \theta^{(k)}) \end{aligned}$$

where $\theta^{(k)}$ are samples from the posterior distribution. We use a Metropolis-Hastings algorithm with a multivariate normal proposal distribution to sample from the corresponding unnormalised log-posterior distribution (1). In our experiments we use a step size of 0.1 and generate 10 randomly-initialised Markov chains of length 1100, in order to avoid the sampler getting stuck in local maxima of spiky, multi-modal posteriors. Discarding the first 100 samples from each chain as burn-in, this leaves 10,000 samples for prediction. (Of course, more elaborate sampling schemes are possible.)

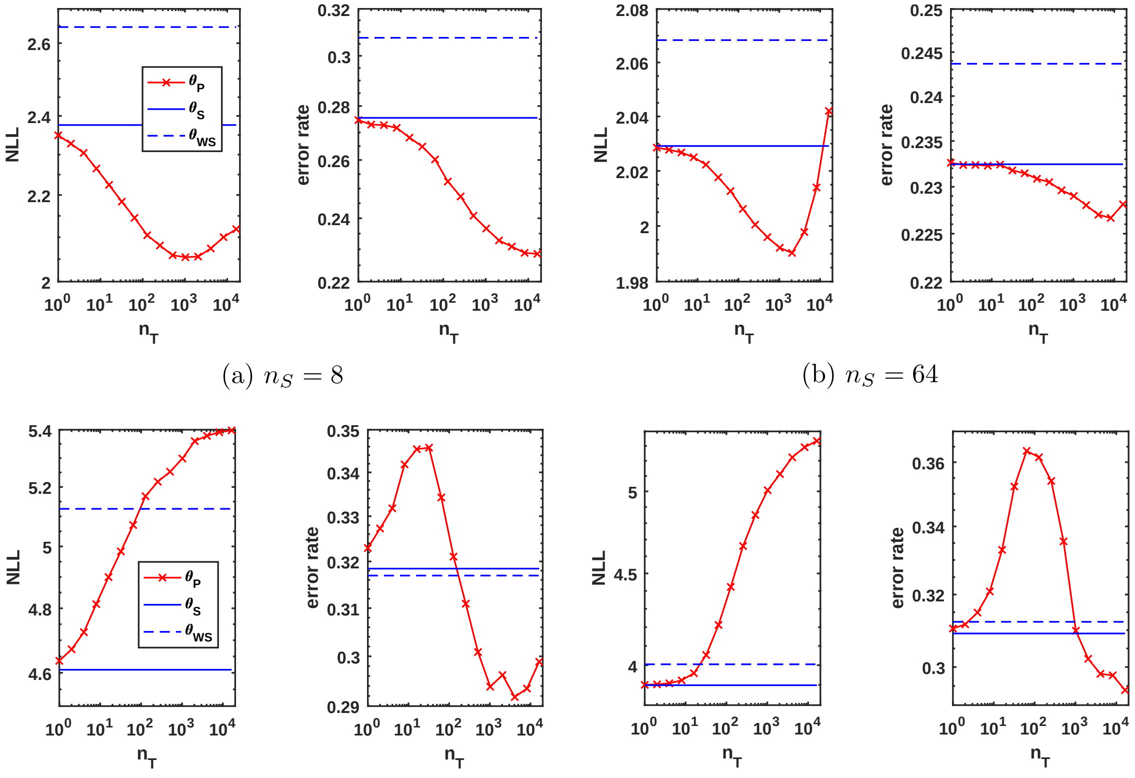
Additionally to the synthetic classification data described in the main paper, we also investigate the setting of model-misspecification. For this, we fit exactly the same model as before (i.e., a linear decision boundary) while changing one of the normal distributions into a mixture of Gaussians (MoG). Specifically, we set $\mu_0 = 0$ and $\mu_1 = 3$ to ensure strong non-linearity and then draw the class-1 effects according to

$$X_E \,|\, (Y = 1) \sim \frac{1}{2}\mathcal{N}(-\mu_1, 1) + \frac{1}{2}\mathcal{N}(\mu_1, 1).$$

Fig. 3 shows the corresponding learning curves for θ_S, θ_{WS}, and θ_{LR}.

Fig. 4 shows two examples of posterior distributions over θ_S, θ_{WS}, and θ_P given labelled and unlabelled training data. For a correct model and given 8 labelled and 1024 unlabelled data (Fig. 4a), the posterior over θ_P, unlike those over θ_S and θ_{WS}, is approximately centred around the true parameter values. Moreover, it is more spiked as indicated by the scaling of axes. Under model-misspecification as shown in Fig. 4b, on the other hand, the posterior over θ_P appears to be bimodal with respect to μ_0 and μ_1, whereas posteriors over θ_S and θ_{WS} seem to remain unimodal.

The decision surfaces, $P(Y = 1 \,|\, X_C, X_E)$, resulting from the different posteriors in Fig. 4b are shown in Fig. 5. It also contains the ground truth as used for generating synthetic model misspecification data . As can be seen, the true decision boundary, $P(Y = 1|X_C, X_E) = 0.5$, is formed by two straight lines separating the"$Y = 0$-cluster" from the Gaussian mixture for



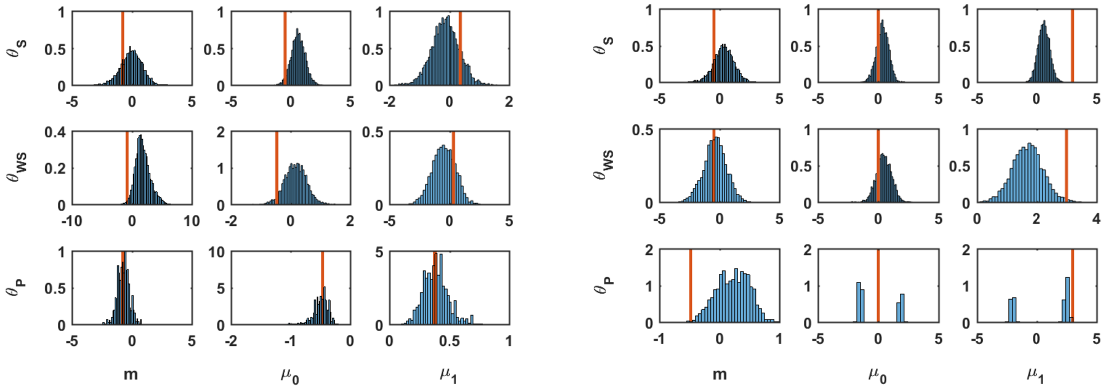
(a) $n_S = 8$

(b) $n_S = 64$

(c) $n_S = 8$ (misspecified)

(d) $n_S = 64$ (misspecified)

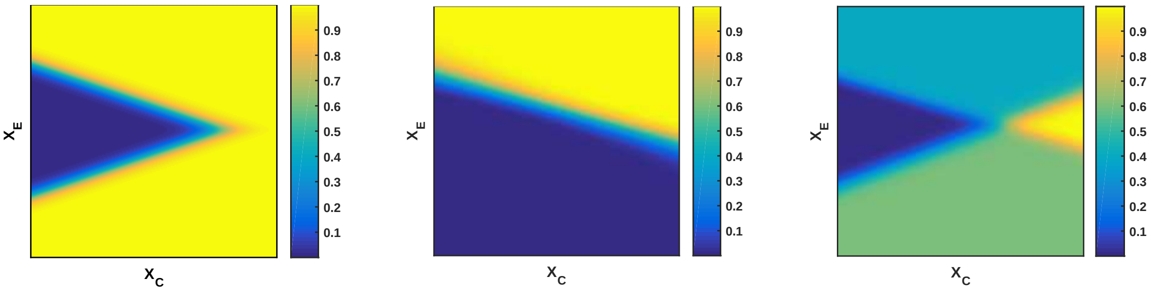
Figure 3: Synthetic classification results using a Bayesian approach for correctly-specified (top row) and misspecified (bottom row) models. Learning curves show test-set averages over 10^3 simulations.



(a) correct model ($n_S = 8$, $n_T = 1024$)

(b) misspecified model ($n_S = 8$, $n_T = 256$)

Figure 4: Metropolis-Hastings sampling-based approximations to the posterior distributions over classification parameters for two example cases of correctly- (a) and incorrectly-specified (b) models; vertical red lines indicate the corresponding true values of the model parameters m, μ_0, and μ_1.



(a) ground truth

(b) using θ_S

(c) using θ_P

Figure 5: Visualization of the probabilistic conditional $P(Y = 1|X_C, X_E)$ under model-misspecification, and its Bayesian approximations with $n_S = 8$, $n_T = 256$, corresponding to the posteriors shown in Fig. 4b. In all three plots, X_C and X_E range from -10 to 10.

$Y = 1$. The decision boundary found using θ_S corresponds to one of these linear segments, whereas that found using θ_P is more differentiated. It appears to be the average of both linear segments taken individually, resulting in class probabilities close to 0.5 over a wide range of (X_C, X_E). This observation is consistent with the bimodal posterior over μ_0 and μ_1 found for θ_P in Fig. 4b, with each mode corresponding to one of the two linear boundaries.

Appendix E. Algorithms

In this Appendix, we provide pseudo-code for training a semi-generative model using a maximum likelihood or a Bayesian approach, and show how to use such a model to make predictions for new data.

Algorithm 1 describes how to train a semi-generative model for a multi-class classification task under unsupervised covariate shift adaptation using maximum likelihood estimation.

Algorithm 1 Semi-generative maximum likelihood estimation for classification

```
Input: labelled source data $\{(x_C^i, y^i, x_E^i)\}_{i=1}^{n_S}$, unlabelled target data $\{(x_C^j, x_E^j)\}_{j=n_S+1}^{n_S+n_T}$, mechanisms $P(Y|X_C, \theta_Y)$ and $P(X_E|Y, \theta_E)$, hyperparameter $\lambda \in (0, 1)$, initial guess $\theta_0$, learning rate $\alpha$
Output: pooled-data, semi-generative MLE $\hat{\theta} = (\hat{\theta}_Y, \hat{\theta}_E)$
$\ell_S(\theta) \leftarrow \sum_{i=1}^{n_S} \log P(y^i|x_C^i, \theta_Y) + \log P(x_E^i|y^i, \theta_E)$
$\ell_T(\theta) \leftarrow \sum_{j=n_S+1}^{n_S+n_T} \log \left( \sum_{y=1}^{k} P(y|x_C^j, \theta_Y) P(x_E^j|y, \theta_E) \right)$
$\ell_P(\theta) \leftarrow \frac{\lambda}{n_S} \ell_S(\theta) + \frac{1-\lambda}{n_T} \ell_T(\theta)$
$t \leftarrow 0$
while not converged do
    $\theta_{t+1} \leftarrow \theta_t + \alpha \nabla_\theta \ell_P(\theta_t)$
    $t \leftarrow t + 1$
end while
$\hat{\theta} \leftarrow \theta_t$
```

Algorithm 2 details how to predict class probabilities from a semi-generative model and parameter estimate $\hat{\theta} = (\hat{\theta}_Y, \hat{\theta}_E)$.

Algorithm 2 Label prediction for classification

```
Input: new observation from target domain $(x_C^{\text{new}}, x_E^{\text{new}})$, meachanisms $P(Y|X_C, \theta_Y)$ and
    $P(X_E|Y, \theta_E)$, parameter estimate $\hat{\theta} = (\hat{\theta}_Y, \hat{\theta}_E)$
Output: label probabilities $p_1, ..., p_k$
$Z = \sum_{y=1}^{k} P(y|x_C^{\text{new}}, \hat{\theta}_Y) P(x_E^{\text{new}}|y, \hat{\theta}_E)$
for $y = 1, ..., k$ do
    $p_y = P(y|x_C^{\text{new}}, \hat{\theta}_Y) P(x_E^{\text{new}}|y, \hat{\theta}_E)/Z$
end for
```

Algorithm 3 describes how to perform Bayesian inference with a semi-generative model. It uses the Metropolis-Hastings algorithm as an example for sampling from the posterior distribution. However, more elaborate sampling approach are, of course, possible.

Algorithm 3 Semi-generative Bayes for classification (Metropolis-Hastings sampling)

```
Input: labelled source data $\{(x_C^i, y^i, x_E^i)\}_{i=1}^{n_S}$, unlabelled target data $\{(x_C^j, x_E^j)\}_{j=n_S+1}^{n_S+n_T}$,
    mechanisms $P(Y|X_C, \theta_Y)$ and $P(X_E|Y, \theta_E)$, prior $\pi(\theta)$, hyperparameter $\lambda \in (0, 1)$,
    proposal distribution $\mathcal{N}(\theta_{t+1}|\theta_t, \sigma^2)$
Output: samples from posterior distribution $\theta^{(k)}$
$\ell_S(\theta) \leftarrow \sum_{i=1}^{n_S} \log P(y^i|x_C^i, \theta_Y) + \log P(x_E^i|y^i, \theta_E)$
$\ell_T(\theta) \leftarrow \sum_{j=n_S+1}^{n_S+n_T} \log \big( \sum_{y=1}^{k} P(y|x_C^j, \theta_Y) P(x_E^j|y, \theta_E) \big)$
$\ell(\theta) \leftarrow \log \pi(\theta) + (n_S + n_T) \big( \frac{\lambda}{n_S} \ell_S(\theta) + \frac{1-\lambda}{n_T} \ell_T(\theta) \big)$
$t \leftarrow 0$
$\theta^{(0)} \leftarrow 0$
while Markov chain not mixed do
    $\theta_{\text{cand}} \leftarrow \mathcal{N}(\theta_{\text{cand}} \,|\, \theta^{(t)}, \sigma^2)$
    $u \leftarrow U(0, 1)$
    if $\exp(\ell(\theta_{\text{cand}}) - \ell(\theta^{(t)})) > u$ then
        $\theta^{(t+1)} \leftarrow \theta_{\text{cand}}$
    else
        $\theta^{(t+1)} \leftarrow \theta^{(t)}$
    end if
    $t \leftarrow t + 1$
end while
```