

Supplementary Material for “**matrans**: an R
package for parameter-transfer learning of
multi-source cancer prognosis data”

Xiaonan Hu^{1,2}, Xinyu Zhang^{3*}, Shuangge Ma⁴

¹School of Mathematics Sciences, Capital Normal University, 105 West
Third Ring Road North, Beijing, 100048, Beijing, China.

²Academy for Multidisciplinary Studies, Capital Normal University, 105
West Third Ring Road North, Beijing, 100048, China.

^{3*}Academy of Mathematics and System Sciences, Chinese Academy of
Sciences, Zhongguancun East Road, Beijing, 100190, Beijing, China.

⁴Department of Biostatistics, Yale School of Public Health, 300 George
Street, New Haven, 06520, CT, USA.

*Corresponding author(s). E-mail(s): xinyu@amss.ac.cn;
Contributing authors: xnhu@amss.ac.cn; shuangge.ma@yale.edu;

S.1 Details of the methodology

We now formally introduce the semiparametric setting summarized in
Section 2.1. Consider a target dataset $\left\{y_i^{(0)}, \boldsymbol{x}_i^{(0)}, \boldsymbol{z}_i^{(0)}\right\}_{i=1}^{n_0}$, where $y_i^{(0)}$ is a
continuous scalar response, $\boldsymbol{x}_i^{(0)} \in \mathbb{R}^p$ denotes the parametric covariates, and
 $\boldsymbol{z}_i^{(0)} \in \mathbb{R}^{q_0}$ denotes the nonparametric covariates. For $m = 1, \dots, M$, let

$\{y_i^{(m)}, \mathbf{x}_i^{(m)}, \mathbf{z}_i^{(m)}\}_{i=1}^{n_m}$ denote the m th source dataset, with $\mathbf{x}_i^{(m)} \in \mathbb{R}^p$ and $\mathbf{z}_i^{(m)} \in \mathbb{R}^{q_m}$. The same p parametric covariates are assumed to be observed across the target and source datasets, whereas the nonparametric covariates may differ across datasets. The partially additive linear model for the m th dataset is

$$y_i^{(m)} = (\mathbf{x}_i^{(m)})^\top \boldsymbol{\beta}^{(m)} + g^{(m)}(\mathbf{z}_i^{(m)}) + \varepsilon_i^{(m)}, \quad (1)$$

where $g^{(m)}(\mathbf{z}_i^{(m)}) = \sum_{l=1}^{q_m} g_l^{(m)}(z_{il}^{(m)})$, and $\varepsilon_i^{(m)}$ has mean zero and variance $\sigma_{i,m}^2$. The coefficient vectors $\boldsymbol{\beta}^{(m)}$ may be identical to or different from $\boldsymbol{\beta}^{(0)}$, allowing the source models to contain different amounts of transferable parametric information.

S.1.1 Optimal model averaging-based parameter-transfer approach (Hu and Zhang 2023)

To estimate the target and source models in (1), we adopt a polynomial spline-based estimator to approximate the nonparametric functions. For the m th model, let $\Psi_l^{(m)}$ denote the polynomial spline space of degree $r_l^{(m)} \geq 1$ and $S_l^{(m)}$ the number of interior knots on $[0, 1]$. Let $B_l^{(m)}(z) = \{b_{l1}(z), \dots, b_{lv_l^{(m)}}(z)\}^\top$ be a normalized B-spline basis, where $v_l^{(m)} = r_l^{(m)} + S_l^{(m)}$. The estimator is then:

$$\hat{\boldsymbol{\theta}}^{(m)} = \arg \min_{\boldsymbol{\theta}^{(m)}} \sum_{i=1}^{n_m} \left\{ y_i^{(m)} - (\mathbf{d}_i^{(m)})^\top \boldsymbol{\theta}^{(m)} \right\}^2, \quad m = 0, \dots, M, \quad (2)$$

with $\mathbf{d}_i^{(m)} = [(\mathbf{x}_i^{(m)})^\top, \{B_1^{(m)}(z_{i1}^{(m)})\}^\top, \dots, \{B_{q_m}^{(m)}(z_{iq_m}^{(m)})\}^\top]^\top$ and $\boldsymbol{\theta}^{(m)} = \{(\boldsymbol{\beta}^{(m)})^\top, (\boldsymbol{\gamma}_1^{(m)})^\top, \dots, (\boldsymbol{\gamma}_{q_m}^{(m)})^\top\}^\top$. For high-dimensional cases where $p \gg$

$\max_m n_m$, we use Lasso estimation:

$$\hat{\boldsymbol{\theta}}^{(m)} = \arg \min_{\boldsymbol{\theta}^{(m)}} \left\{ \frac{1}{2n_m} \sum_{i=1}^{n_m} \left(y_i^{(m)} - (\mathbf{d}_i^{(m)})^\top \boldsymbol{\theta}^{(m)} \right)^2 + \lambda^{(m)} \|\boldsymbol{\beta}^{(m)}\|_1 \right\}.$$

To borrow parameter information from auxiliary source domains, we construct candidate estimators for the target model using coefficient estimates from all $M + 1$ models (target $m = 0$ and sources $m = 1, \dots, M$):

$$\hat{\mu}_{i,m}^{(0)} = (\mathbf{d}_i^{(0)})^\top \hat{\boldsymbol{\theta}}_m^{(0)} = (\mathbf{x}_i^{(0)})^\top \hat{\boldsymbol{\beta}}^{(m)} + \sum_{l=1}^{q_0} \{B_l^{(0)}(z_{il}^{(0)})\}^\top \hat{\boldsymbol{\gamma}}_l^{(0)}, \quad m = 0, \dots, M, \quad (3)$$

with $\hat{\boldsymbol{\theta}}_m^{(0)} = \{(\hat{\boldsymbol{\beta}}^{(m)})^\top, (\hat{\boldsymbol{\gamma}}_1^{(0)})^\top, \dots, (\hat{\boldsymbol{\gamma}}_{q_0}^{(0)})^\top\}^\top$. The final transfer learning prediction is $\hat{\mu}_i^{(0)}(\mathbf{w}) = \sum_{m=0}^M w_m \hat{\mu}_{i,m}^{(0)}$, where $\mathbf{w} = (w_0, \dots, w_M)^\top$ lies in $\mathcal{W} = \{\mathbf{w} \in [0, 1]^{M+1} : \sum_{m=0}^M w_m = 1\}$.

Unlike conventional source-selection procedures, our algorithm does not first classify the source datasets as transferable or non-transferable. Instead, all target- and source-based estimators are retained as candidates, and their contributions are determined through nonnegative data-driven weights. Source relevance is therefore represented continuously through the estimated weights, avoiding a separate binary source-screening step and the specification of a source-similarity threshold.

S.1.2 Details of the parameter-transfer algorithm

The `matrans` package implements the Trans-SMAP algorithm of [Hu and Zhang \(2023\)](#). We obtain optimal weights via J -fold cross-validation ($J > 1$). The target samples are randomly partitioned into J disjoint subgroups $\mathcal{G}_1, \dots, \mathcal{G}_J$,

with complements $\mathcal{G}_j^c = \{1, \dots, n_0\} \setminus \mathcal{G}_j$. The cross-validation criterion is:

$$CV(\mathbf{w}) = \frac{1}{n_0} \sum_{j=1}^J \sum_{i \in \mathcal{G}_j} \left(y_i^{(0)} - \hat{\mu}_{i, [\mathcal{G}_j^c]}^{(0)}(\mathbf{w}) \right)^2, \quad (4)$$

where $\hat{\mu}_{i, [\mathcal{G}_j^c]}^{(0)}(\mathbf{w})$ is the weighted average of $\hat{\mu}_{i, m, [\mathcal{G}_j^c]}^{(0)}$, defined analogously to $\hat{\mu}_{i, m}^{(0)}$ but using only $\mathcal{G}_j^c$. The optimal weight vector minimizes $CV(\mathbf{w})$ over $\mathbf{w} \in \mathcal{W}$:

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w} \in \mathcal{W}} CV(\mathbf{w}).$$

The algorithm is summarized below.

Algorithm 1: Trans-SMAP algorithm

Input: Training data $\{(\mathbf{x}_i^{(m)}, \mathbf{z}_i^{(m)}, y_i^{(m)})\}_{i=1}^{n_m}$ for $m = 0, \dots, M$ from (1), and
a new target sample $(\mathbf{x}_{n_0+1}^{(0)}, \mathbf{z}_{n_0+1}^{(0)})$.

Output: Prediction $\hat{y}_{n_0+1}^{(0)}$.

1. Estimate $\hat{\boldsymbol{\theta}}^{(m)}$ for $m = 0, \dots, M$ using (2) on all training data.
 2. Randomly split target training samples into J disjoint subgroups
($2 \leq J \leq n_0$).
 3. **foreach** $j = 1, \dots, J$ **do**
 - (a) For $m = 1, \dots, M$, estimate $\boldsymbol{\theta}^{(m)}$ via (2) using all data; estimate $\boldsymbol{\theta}^{(0)}$ using only $\mathcal{G}_j^c$.
 - (b) For $i \in \mathcal{G}_j$, $m = 0, \dots, M$, compute $\hat{\mu}_{i, m, [\mathcal{G}_j^c]}^{(0)}$ via (3).
 - (c) Form $\hat{\mu}_{i, [\mathcal{G}_j^c]}^{(0)}(\mathbf{w}) = \sum_{m=0}^M w_m \hat{\mu}_{i, m, [\mathcal{G}_j^c]}^{(0)}$.
 4. Minimize CV criterion (4) to obtain optimal $\hat{\mathbf{w}}$.
 5. Output $\hat{\mu}_{n_0+1}^{(0)}(\hat{\mathbf{w}}) = \sum_{m=0}^M \hat{w}_m \hat{\mu}_{n_0+1, m}^{(0)}$ for the new sample.
-

To estimate the $M + 1$ semiparametric models, we use B-splines from the `splines` package, with the number of knots specified via the `df` argument in `bs`. Define $\mathbf{Q} = n_0^{-1} \sum_{j=1}^J \sum_{i \in \mathcal{G}_j} (y_i^{(0)} \mathbf{1} - \hat{\mathbf{y}}_{i, [\mathcal{G}_j^c]})(y_i^{(0)} \mathbf{1} - \hat{\mathbf{y}}_{i, [\mathcal{G}_j^c]})^\top$, where

$\hat{\mathbf{y}}_{i, [\mathcal{G}_j^c]} = (\hat{\mu}_{i,0, [\mathcal{G}_j^c]}^{(0)}, \dots, \hat{\mu}_{i,M, [\mathcal{G}_j^c]}^{(0)})^\top$ and $\mathbf{1}$ is an $(M+1)$ -vector of ones. Under $\sum_{m=0}^M w_m = 1$, we have

$$\begin{aligned}
CV(\mathbf{w}) &= \frac{1}{n_0} \sum_{j=1}^J \sum_{i \in \mathcal{G}_j} \left\{ y_i^{(0)} - \hat{\mu}_{i, [\mathcal{G}_j^c]}^{(0)}(\mathbf{w}) \right\}^2 \\
&= \frac{1}{n_0} \sum_{j=1}^J \sum_{i \in \mathcal{G}_j} \left\{ \mathbf{w}^\top (y_i^{(0)} \mathbf{1} - \hat{\mathbf{y}}_{i, [\mathcal{G}_j^c]}) \right\}^2 \\
&= \mathbf{w}^\top \left[\frac{1}{n_0} \sum_{j=1}^J \sum_{i \in \mathcal{G}_j} \left\{ (y_i^{(0)} \mathbf{1} - \hat{\mathbf{y}}_{i, [\mathcal{G}_j^c]})(y_i^{(0)} \mathbf{1} - \hat{\mathbf{y}}_{i, [\mathcal{G}_j^c]})^\top \right\} \right] \mathbf{w} \\
&= \mathbf{w}^\top \mathbf{Q} \mathbf{w}.
\end{aligned}$$

Hence, weight estimation reduces to a constrained quadratic programming problem, efficiently solved via `solve.QP` in the `quadprog` package.

S.2 Simulation studies

To provide a more comprehensive assessment of the package, we consider two additional groups of simulation experiments under high-dimensional linear models. The first group varies the degree of parameter similarity between the target and source models, thereby controlling the transferability of the available source information. The second group varies the error variance to investigate the effect of the signal-to-noise level. For both groups, we consider correctly specified and misspecified model settings. We compared the proposed method with Trans-SimpMA; transfer learning via smoothed-AIC and smoothed-BIC (?) model averaging (Trans-SAIC and Trans-SBIC); Lasso-Tar; Lasso-All; Trans-Lasso; and Trans-GLM. To evaluate performance, we

generate $n^* = 100$ test samples from the target model and compute MSPE: $\text{MSPE} = \frac{1}{n^*} \sum_{i=1}^{n^*} (\hat{\mu}_i^{(0)} - \mu_i^{(0)})^2$, and SSE: $\text{SSE} = \|\hat{\beta}^{(0)} - \beta^{(0)}\|^2$. All competing methods are evaluated using the same simulated datasets, tuning procedures, and test samples, with SSE and MSPE used to measure estimation and prediction performance, respectively.

We set $p = 500$, $M = 6$, and $(n_0, \dots, n_M) = (100, 200, 150, 100, 100, 150, 100)$. Under parametric settings, $\mathbf{x}_i^{(m)} \sim N(\mathbf{0}, \Sigma)$ with $\Sigma_{aa'} = 0.5^{|a-a'|}$. The random errors $\varepsilon_i^{(m)} \sim N(0, 1)$. This simulation method is termed Trans-HMAP. Tuning parameters $\lambda^{(m)}$ are selected by 8-fold CV. The target coefficient vector is $\beta^{(0)} = (0.5 \cdot \mathbf{1}_5^\top, \mathbf{0}_{495}^\top)^\top$. In a misspecification scenario, target and source 6 have an extra coefficient $\beta_{p+1} = 0.5$, omitted during fitting.

S.2.1 Performance under different levels of parameter similarity

We first investigate the effect of source–target similarity on transfer learning. Source coefficients are:

- $m = 1, 2, 3, 5$: $\beta_j^{(m)} = \beta_j^{(0)} + \delta \psi_j \mathbb{I}(j \in \{6, \dots, 25\})$;
- $m = 6$: $\beta_j^{(6)} = \beta_j^{(0)}$;
- $m = 4$: $\beta_j^{(4)} = \beta_j^{(0)} + 0.2 \mathbb{I}(j \in \{1, \dots, 50\})$,

with $\psi_j \in \{-1, 1\}$ equiprobably, where $\delta \in \{0.2, 0.8, 1.5\}$ controls the differences between the source and target coefficient vectors. Smaller values correspond to source models that are more similar to the target model, whereas larger values represent greater source–target heterogeneity. Thus, sources 4

and 6 are constantly informative. For each similarity level, we consider both correctly specified and misspecified models. All results are averaged over 500 independent replications.

Table S1 summarizes the estimation, prediction results, and computational time under different levels of source–target parameter similarity. As δ increases, the SSE and MSPE of Trans-HMAP increase only moderately, whereas several procedures that borrow source information more aggressively, particularly Lasso-All, Trans-SimpMA, Trans-SBIC, and Trans-SAIC, deteriorate substantially. This pattern illustrates the importance of adaptively controlling the contribution of heterogeneous source information. The uplift rate is calculated relative to the target-only estimator Lasso-Tar. Under correct model specification, Trans-HMAP reduces SSE by 46.8%, 25.9%, and 22.5% for $\delta = 0.2, 0.8, 1.5$, respectively. The corresponding reductions in MSPE are 42.8%, 25.6%, and 22.9%. Under model misspecification, the SSE reductions are 54.0%, 27.6%, and 24.9%, while the corresponding MSPE reductions are 28.0%, 16.5%, and 14.3%. Thus, Trans-HMAP consistently improves upon target-only estimation and prediction across all similarity levels considered, although the relative benefit generally decreases as the source–target discrepancy increases.

When $\delta = 0.2$, the source and target models are highly similar, and direct pooling or more aggressive transfer procedures can achieve smaller errors than Trans-HMAP. At the intermediate similarity level $\delta = 0.8$, Trans-HMAP attains the smallest SSE and MSPE under both correct specification and misspecification. When $\delta = 1.5$, Trans-GLM achieves slightly smaller errors, but Trans-HMAP remains competitive and substantially outperforms Lasso-Tar,

Table S1 Averaged SSE, MSPE, and computation time (in seconds) for different methods under varying source–target similarity parameters δ . The upper and lower panels correspond to correct specification and model misspecification, respectively. Standard errors are shown in parentheses

Methods	$\delta = 0.2$			$\delta = 0.8$			$\delta = 1.5$		
	SSE	MSPE	Time	SSE	MSPE	Time	SSE	MSPE	Time
Trans-HMAP	0.123 (0.040)	0.159 (0.053)	2.644	0.172 (0.065)	0.212 (0.076)	2.717	0.179 (0.066)	0.215 (0.076)	2.737
Trans-SimpMA	0.124 (0.029)	0.175 (0.051)	0.000	0.628 (0.154)	0.840 (0.320)	0.000	2.316 (0.627)	2.842 (1.136)	0.000
Trans-SBIC	0.109 (0.028)	0.160 (0.048)	0.001	0.395 (0.104)	0.535 (0.205)	0.001	1.258 (0.389)	1.537 (0.662)	0.001
Trans-SAIC	0.152 (0.056)	0.213 (0.084)	0.001	0.650 (0.191)	0.853 (0.357)	0.000	2.502 (0.815)	2.984 (1.328)	0.000
Lasso-Tar	0.231 (0.121)	0.278 (0.137)	0.000	0.232 (0.121)	0.285 (0.138)	0.000	0.231 (0.121)	0.279 (0.137)	0.000
Lasso-All	0.074 (0.034)	0.081 (0.042)	0.306	1.091 (0.315)	1.271 (0.504)	0.498	6.675 (1.483)	7.078 (2.155)	0.649
Trans-Lasso	0.086 (0.069)	0.096 (0.071)	0.252	0.301 (0.231)	0.312 (0.223)	0.253	1.234 (1.282)	1.088 (1.023)	0.273
Trans-GLM	0.093 (0.081)	0.120 (0.087)	4.952	0.204 (0.122)	0.220 (0.134)	5.011	0.165 (0.095)	0.174 (0.097)	4.727
Uplift Rate (%)	46.8%	42.8%		25.9%	25.6%		22.5%	22.9%	
Trans-HMAP	0.139 (0.048)	0.424 (0.082)	2.610	0.215 (0.078)	0.496 (0.105)	2.705	0.226 (0.089)	0.505 (0.117)	2.718
Trans-SimpMA	0.130 (0.027)	0.430 (0.074)	0.000	0.638 (0.157)	1.116 (0.350)	0.000	2.347 (0.651)	3.075 (1.127)	0.000
Trans-SBIC	0.118 (0.028)	0.418 (0.073)	0.000	0.436 (0.112)	0.841 (0.237)	0.000	1.426 (0.437)	1.948 (0.727)	0.000
Trans-SAIC	0.164 (0.055)	0.473 (0.102)	0.000	0.684 (0.197)	1.154 (0.389)	0.000	2.716 (0.911)	3.427 (1.392)	0.000
Lasso-Tar	0.302 (0.148)	0.589 (0.179)	0.000	0.297 (0.158)	0.594 (0.192)	0.000	0.301 (0.148)	0.589 (0.179)	0.000
Lasso-All	0.078 (0.037)	0.335 (0.066)	0.311	1.104 (0.317)	1.551 (0.558)	0.502	6.809 (1.580)	7.335 (2.235)	0.650
Trans-Lasso	0.089 (0.074)	0.351 (0.091)	0.248	0.337 (0.267)	0.577 (0.234)	0.248	1.196 (1.194)	1.311 (1.025)	0.265
Trans-GLM	0.115 (0.108)	0.383 (0.113)	4.972	0.265 (0.160)	0.530 (0.187)	5.069	0.206 (0.131)	0.463 (0.152)	4.729
Uplift Rate (%)	54.0%	28.0%		27.6%	16.5%		24.9%	14.3%	

Lasso-All, and Trans-SimpMA. Therefore, the results support the robustness of Trans-HMAP across different levels of source transferability, rather than uniform dominance over all competing methods. The weight patterns in Fig. S1 are consistent with this interpretation: greater weights are assigned to the target model and the more informative source models, whereas less compatible source models receive smaller contributions as source–target heterogeneity increases. The average computation time of Trans-HMAP remains stable at approximately 2.6–2.7 seconds per replication across the settings considered. Although it is slower than Trans-Lasso, it is consistently faster than Trans-GLM.

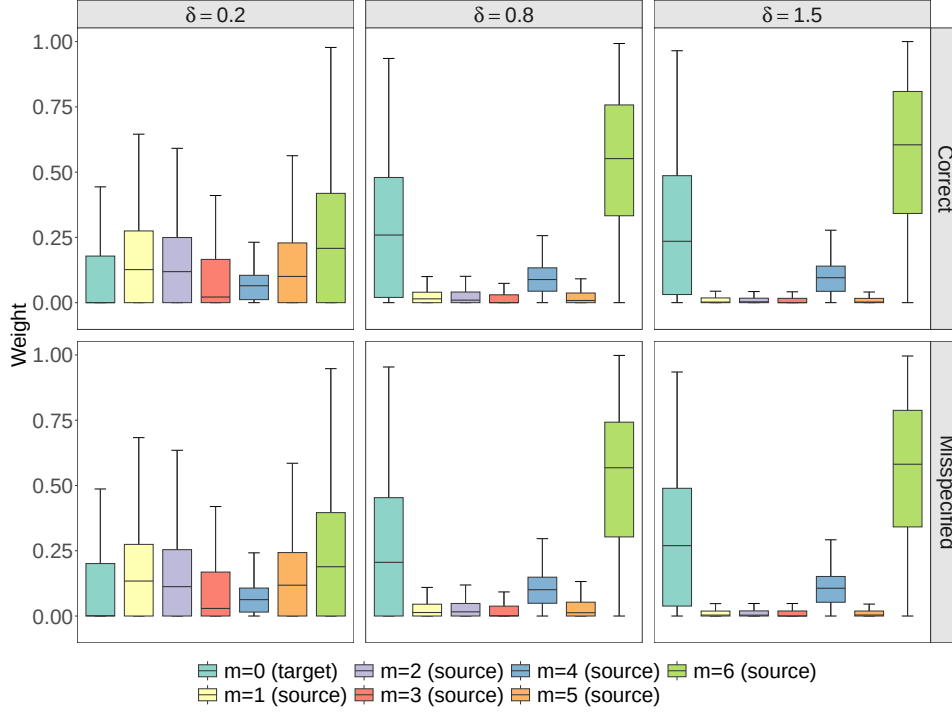


Fig. S1 Weight assignments of Trans-HMAP for target and source models based on 500 replications

S.2.2 Performance under different noise levels

We next examine the sensitivity of the competing methods to the noise level. The model coefficients, covariate distributions, and sample sizes are held fixed, while the error standard deviation is varied over $\{1, 2, 5\}$. Correctly specified and misspecified models are considered separately, and the results are evaluated using SSE and MSPE over 500 replications. Table S2 reports the estimation, prediction results and computational time under different error standard deviations. As expected, the SSE and MSPE values generally increase with σ , reflecting the greater difficulty of estimation and prediction under lower signal-to-noise ratios. Under correct model specification, Trans-HMAP reduces SSE relative to Lasso-Tar by 24.7%, 33.4%, and 63.8% for $\sigma = 1, 2$, and

5, respectively. The corresponding reductions in MSPE are 23.7%, 32.3%, and 59.4%. Under model misspecification, the SSE reductions are 28.5%, 33.2%, and 63.1%, while the corresponding MSPE reductions are 15.6%, 25.6%, and 57.5%. These results indicate that incorporating source information substantially improves upon target-only estimation across all noise levels considered, under both correct specification and misspecification. In particular, the relative improvement over Lasso-Tar becomes more pronounced as the noise level increases. Trans-HMAP achieves the smallest or nearly smallest estimation and prediction errors under low and moderate noise levels. At $\sigma = 5$, some competing transfer learning or model averaging methods attain smaller SSE or MSPE values, although Trans-HMAP continues to provide substantial improvements over the target-only estimator. Thus, the results support the robustness of the proposed method relative to target-only learning, rather than uniform dominance over every competing method. The average computation time of Trans-HMAP ranges from approximately 2.8 to 3.2 seconds per replication across the settings considered. It is more computationally demanding than Trans-Lasso but consistently faster than Trans-GLM, whose average computation time ranges from approximately 5.2 to 6.1 seconds.

Table S2 Averaged SSE, MSPE, and computation time (in seconds) for different methods under varying error standard deviations σ . The upper and lower panels correspond to correct specification and model misspecification, respectively. Standard errors are shown in parentheses

Methods	$\sigma = 1$			$\sigma = 2$			$\sigma = 5$		
	SSE	MSPE	Time	SSE	MSPE	Time	SSE	MSPE	Time
Trans-HMAP	0.174 (0.062)	0.212 (0.073)	2.872	0.540 (0.172)	0.684 (0.246)	3.133	2.751 (0.649)	3.521 (1.022)	3.192
Trans-SimpMA	0.621 (0.157)	0.833 (0.312)	0.000	0.678 (0.146)	0.984 (0.310)	0.000	1.812 (0.215)	2.577 (0.697)	0.000
Trans-SBIC	0.388 (0.103)	0.527 (0.195)	0.001	0.581 (0.141)	0.850 (0.273)	0.001	1.884 (0.459)	2.644 (0.806)	0.000
Trans-SAIC	0.637 (0.190)	0.841 (0.338)	0.000	0.746 (0.294)	1.041 (0.419)	0.000	2.022 (0.822)	2.794 (1.080)	0.000
Lasso-Tar	0.231 (0.121)	0.278 (0.137)	0.000	0.811 (0.381)	1.011 (0.490)	0.000	7.596 (2.006)	8.662 (2.590)	0.000
Lasso-All	1.075 (0.311)	1.249 (0.481)	0.528	1.439 (0.356)	1.618 (0.546)	0.545	5.899 (0.869)	5.671 (1.178)	0.592
Trans-Lasso	0.297 (0.229)	0.298 (0.199)	0.273	0.600 (0.366)	0.674 (0.405)	0.254	5.271 (1.900)	5.602 (1.823)	0.176
Trans-GLM	0.204 (0.121)	0.219 (0.132)	5.260	0.564 (0.382)	0.704 (0.425)	6.056	1.624 (1.787)	2.520 (1.962)	6.013
Uplift Rate (%)	24.7%	23.7%		33.4%	32.3%		63.8%	59.4%	
Trans-HMAP	0.216 (0.081)	0.497 (0.111)	2.762	0.577 (0.184)	0.966 (0.254)	3.033	2.785 (0.674)	3.739 (1.019)	2.908
Trans-SimpMA	0.635 (0.159)	1.077 (0.312)	0.000	0.684 (0.147)	1.255 (0.340)	0.000	1.819 (0.235)	2.797 (0.673)	0.000
Trans-SBIC	0.436 (0.111)	0.819 (0.219)	0.000	0.595 (0.136)	1.126 (0.289)	0.000	1.895 (0.456)	2.869 (0.792)	0.000
Trans-SAIC	0.682 (0.206)	1.118 (0.361)	0.000	0.749 (0.273)	1.307 (0.427)	0.000	2.032 (0.822)	3.024 (1.107)	0.000
Lasso-Tar	0.302 (0.148)	0.589 (0.179)	0.000	0.864 (0.397)	1.298 (0.500)	0.000	7.557 (1.766)	8.793 (2.300)	0.000
Lasso-All	1.108 (0.339)	1.518 (0.523)	0.524	1.455 (0.349)	1.898 (0.603)	0.544	5.980 (0.898)	5.962 (1.193)	0.518
Trans-Lasso	0.322 (0.238)	0.575 (0.239)	0.256	0.629 (0.392)	0.963 (0.404)	0.246	5.157 (1.609)	5.770 (1.937)	0.164
Trans-GLM	0.273 (0.166)	0.533 (0.189)	5.248	0.565 (0.346)	0.972 (0.406)	5.996	1.700 (1.902)	2.835 (1.963)	5.386
Uplift Rate (%)	28.5%	15.6%		33.2%	25.6%		63.1%	57.5%	