Supplementary Materials to: Improved estimation of autoregressive models through contextual impulses and robust modeling

Janne K. Adolf[1] & Eva Ceulemans[1]

[1] KU Leuven - University of Leuven

Abstract

The materials in this document supplement the manuscript *Improved estimation of autoregressive models through contextual impulses and robust modeling*, which is available as a preprint on PsyArXiv. The supplementary materials in this document are divided into materials that concern the first, concepual part of the paper, materials that concern the simulation study, and materials supplementing the empirical illustration section. A clickable table of contents allows to quickly navigate to the specific materials of interest.

Janne K. Adolf & Eva Ceulemans, Research Group of Quantitative Psychology and Individual Differences, Faculty of Psychology and Educational Sciences, KU Leuven - University of Leuven, Belgium.

Additional supplementary materials next to this document are: the *R* package *simARO* enabling *R*-level reproducibility of the raw simulation results, the *use_simARO R*-project supporting the usage of the *simARO* package, the raw simulation results, and the analysis code for the empirical illustrations. All of these components are accessible via the Open Science Framework (OSF) repository *Supplementary materials for manuscript: Improved estimation of autoregressive models through contextual impulses and robust modeling* (Adolf & Ceulemans, 2023); The data used for the empirical illustrations can be found in the Open Science Framework repositories *One does not simply correct for serial dependence: Supplementary materials* (Ariens, 2023).

This work was supported by a research fellowship from the German Research Foundation (DFG) awarded to Janne Adolf, and by research grants from the Fund for Scientific Research-Flanders (FWO, Project No. G.074319N) and the Research Council of KU Leuven (C14/19/054 and iBOF/21/090) awarded to Eva Ceulemans. Some resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government.

Excerpts (of earlier versions) of this work were presented at the 2021 Society for Ambulatory Assessment Conference in (virtual) Zurich, Switzerland, the 2022 International Meeting of the Psychometric Society in Bologna, Italy, the 2023 Society for Ambulatory Assessment Conference in Amsterdam, the Netherlands, and the 2023 10th European Congress of Methodology in Gent, Belgium.

The authors would like to thank Ruben Crevits and Christophe Croux for sharing their code, and Kristof Meers for his support with using the VSC and setting up the GitLab infrastructure accompanying this work. This document was created using R Markdown and papaja (Aust & Barth, 2022).

Correspondence concerning this article should be addressed to Janne K. Adolf, Tiensestraat 102, 3000 Leuven, Belgium. E-mail: janne.adolf@kuleuven.be

Contents

1 Supplementary materials to the presentation on autoregressive systems

1.1 Behavior of autoregressive systems with and without outliers

Figure 1 in the manuscript *Improved estimation of autoregressive models through contextual impulses and robust modeling* demonstrates the behavior of a first-order autoregressive (AR(1)) system in the absence and presence of outliers. Animated variants of the figure's panels were included in the most recent presentation on the project given at the International Meeting of the Psychometric Society 2022 in Bologna, Italy. The slides are available via the Open Science Framework (link).

1.2 Robust state space modeling

1.2.1 Robust maximum likelihood estimation

Classical maximum likelihood estimation. State space modeling (SSM) relies on the prediction error decomposition-form of the log-likelihood function (Harvey, 1989, p. 126, Eq. 3.4.6). For the sample $y_1,\ y_2, ..., y_t, ..., y_T$ the prediction error decomposition log-likelihood is calculated as:

$$\ell\left(\boldsymbol{\theta}; y_1,\ y_2, ..., y_t, ..., y_T\right) = -\frac{1}{2}\left(p\,T\log(2\pi) + \sum_{t=1}^{T}\left(\log(|F_t|) + {v_t}'\,F_t^{-1}\,v_t\right)\right), \tag{1}$$

where $\boldsymbol{\theta}$ is the vector of model parameters, p is the number of observed variables, $v_t\,;\,t = 1, 2, ..., T$ are $p \times 1$ Kalman filter (KF)-generated prediction errors and $F_t\,;\,t = 1, 2, ..., T$ their $p \times p$ covariance matrices.

Crevits and Croux (2019) present a variant, $-\ell^*$, that is the negative log-likelihood divided by T minus the constant term $-\frac{p\log(2\pi)}{2}$:

$$-\ell^*\left(\boldsymbol{\theta}; y_1,\ y_2, ..., y_t, ..., y_T\right) = \frac{1}{2\,T}\left(\sum_{t=1}^{T}\left(\log(|F_t|) + {v_t}'\,F_t^{-1}\,v_t\right)\right). \tag{2}$$

Robust maximum likelihood estimation. Based on $-\ell^*$, Crevits & Croux present their maximum trimmed likelihood (MTL) estimator, which trims off the α-fraction of data points that are the least likely under the model. To this end, the Mahalanobis distances between the data and the KF-generated, model-implied predictions are calculated for each data point as:

$$d_t\left(\boldsymbol{\theta}\right) = {v_t}'\,F_t^{-1}\,v_t. \tag{3}$$

Then the subset D of time points is determined that correspond to the $(1-\alpha)$ fraction (or $(1-\alpha) * 100$ %) of data points with the lowest distances. Only these data points are considered in the trimmed negative log-likelihood function:

$$-\ell^*_{trim}\left(\boldsymbol{\theta}; y_{t\,\in D}\right) = \frac{1}{2\,T\,(1-\alpha)}\left(\sum_{t\in D}\left(\log(|F_t|) + c_{trim}\left({v_t}'\,F_t^{-1}\,v_t\right)\right)\right). \tag{4}$$

The constant c_{trim} is introduced to make "the expected value of the trimmed log-likelihood equal to that of the untrimmed log-likelihood at the true model parameter" (Crevits & Croux, 2019,

p. 1697). It is calculated as:

$$c_{trim} = \frac{1}{F_{\chi^2_{p+2}}(F^{-1}_{\chi^2_p}(1-\alpha))}, \tag{5}$$

where $F_{\chi^2_{p+2}}(\cdot)$ is the χ^2 distribution function with $df = p+2$, and $F^{-1}_{\chi^2_p}(\cdot)$ is the χ^2 quantile function with $df = p$. If $\alpha = 0$, then all observations are used for estimation and $c_{trim} = 1$, so the log-likelihood function simplifies to the classical one, $-\ell^*$.

1.2.2 Robust Kalman filter

Classical Kalman filter. In SSM the (classical) KF serves parameter estimation as it generates input for the prediction error decomposition log-likelihood function: the prediction errors $v_t\,;\, t = 1, 2, ..., T$ and their covariance matrices $F_t\,;\, t = 1, 2, ..., T$ (see previous section). These quantities are calculated when the KF is used with its original purpose, estimating the states of a latent process from noisy observations over time. To this end the familiar dynamic system equations are used:

$$\underset{q\times 1}{\eta_t} = \underset{q\times 1}{\alpha} + \underset{q\times q}{B}\,\underset{q\times 1}{\eta_{t-1}} + \underset{q\times 1}{\zeta_t} \quad \text{(transition equation)} \tag{6}$$

$$\underset{p\times 1}{y_t} = \underset{p\times 1}{\tau} + \underset{p\times q}{\Lambda}\,\underset{q\times 1}{\eta_t} + \underset{p\times 1}{\varepsilon_t} \quad \text{(observation equation)} \tag{7}$$

with

$$\zeta_t \overset{iid}{\sim} N(0, \Psi) \quad \text{and} \quad \varepsilon_t \overset{iid}{\sim} N(0, \Theta) \quad \text{for} \quad t \in T$$

In 'traditional' KF applications, the parameters that determine the dynamics of the latent and observed processes are assumed to be known. If the KF is used for parameter estimation (current) parameter estimates are used to in turn estimate the latent process states (and subsequently calculate the likelihood given the parameter values).

To arrive at latent process state estimates, the *classical KF* takes two steps, a prediction step and an updating step. To understand the *prediction step*, let's assume we have an estimate of the latent process state at time $t-1$, η_{t-1}, which is denoted as $\hat{\eta}_{t-1|t-1}$. The associated estimation error is $e_{t-1|t-1} := \eta_{t-1} - \hat{\eta}_{t-1|t-1}$ with covariance matrix $P_{t-1|t-1}$.

Based on this, the KF predicts a *process state* for time t, $\hat{\eta}_{t|t-1}$ [1], using the above transition equation:

$$\hat{\eta}_{t|t-1} = \alpha + B\,\hat{\eta}_{t-1|t-1} \quad \text{(state prediction)} \tag{8}$$

The covariance matrix of the estimation error $e_{t|t-1} := \eta_t - \hat{\eta}_{t|t-1}$, $P_{t|t-1}$ [2], is

$$P_{t|t-1} = B\,P_{t-1|t-1}\,B^t + \Psi \quad \text{(error cov.mat. prediction)} \tag{9}$$

The KF can now predict the *observation* at time t via the observation equation and, given the observation y_t, calculate the prediction error v_t:

$$\hat{y}_{t|t-1} = \tau + \Lambda\,\hat{\eta}_{t|t-1} \quad \text{(observation prediction)} \tag{10}$$

$$v_t = y_t - \hat{y}_{t|t-1} \quad \text{(prediction error)} \tag{11}$$

[1] If (and only if) there is no measurement error, this will be the conditional expectation of the process, so the expected value of the process given the (true) process state at $t-1$ (go to the description of the *updating step* to see why). This holds for all $t > 1$ independent of how the KF was initialized (see the description of the *initialization*).

[2] If (and only if) there is no measurement error, this will be the conditional covariance matrix of the process, so the covariance matrix given the (true!) process state at $t-1$, *Psi* (go to the description of the *updating step* to see why). This holds for all $t > 1$ independent of how the KF was initialized (see the description of the *initialization*).

The prediction error v_t has covariance matrix

$$F_t = \Lambda\, P_{t|t-1}\, \Lambda^t + \Theta \quad \text{(cov.mat. prediction error)} \tag{12}$$

which can be derived by re-writing v_t as $v_t = \Lambda\, e_{t|t-1} + \varepsilon_t$.

Now follows the *updating step* during which the KF corrects the state estimate generated during the prediction step by taking the prediction error at time t into account. The updated state estimate $\hat{\eta}_{t|t}$ [3] is

$$\hat{\eta}_{t|t} = \hat{\eta}_{t|t-1} + K_t\, v_t \quad \text{(state update)} \tag{13}$$

with Kalman gain matrix K_t

$$K_t = P_{t|t-1}\, \Lambda^t\, F_t^{-1} \quad \text{(Kalman gain mat.)} \tag{14}$$

The covariance matrix of the estimation error $e_{t|t} := \eta_t - \hat{\eta}_{t|t}$, $P_{t|t}$ [4], is

$$P_{t|t} = P_{t|t-1} - K\, \Lambda\, P_{t|t-1} \quad \text{(error cov.mat. update)} \tag{15}$$

The updating depends on the amount of measurement error variance the KF expects: A larger measurement error variance means that less weight is given to the prediction errors and thus actual observations. The observations are thus treated as less trustworthy and the KF relies more on its own predictions. In turn, if the measurement error variance is expected to be small, more weight os given to the actual observations relative to predicted states. If no measurement error effects are expected, the KF always updates state estimates to the observed states (i.e., the data are fully trusted).

Before the KF can take it's first prediction step, it needs to be *initialized*. That is, to generate a state prediction for time $t = 1$, the KF needs starting values, which are values for $\hat{\eta}_{0|0}$ and $P_{0|0}$. If these are set to the true, unconditional moments of the latent process, then the state prediction for $t = 1$ will be the unconditional mean and the covariance matrix of the estimation error will be the unconditional covariance matrix of the process. This might matter for parameter estimation, where the KF starting values might be constrained to the unconditional process moments as a function of the model parameters. Another strategy is to use the observed moments of the process for initialization.

Robust Kalman filter. So far we were concerned with the classical KF, now we turn to the *robustified KF* as presented in Crevits and Croux (2019). In a nutshell, the robustification adapts the measurement error-related updating step of the KF: The impact of unexpectedly extreme observations - which come with large prediction errors - is damped overproportionally. This directly affects the estimation of process states, but also parameter estimation since both processes are inseparably intertwined. A robustification parameter denoted k, which determines the additional damping, is to be set a priori to applying the robust KF.

In terms of equations, the robust KF relies on the above, except for an adapted covariance matrix of the prediction error, which looks like this:

$$F_t^r = \Lambda\, P_{t|t-1}\, \Lambda^t + \Theta^{\frac{1}{2}}\, W_t^{-1}\, \Theta^{\frac{1}{2}} \quad \text{(cov.mat. prediction error, robustified)} \tag{16}$$

[3] If (and only if) there is no measurement error (i.e., $\Theta = 0$), then $K = \Lambda^{-1}$, and the state update becomes $\Lambda\,(y_t - \tau)$ and $\hat{\eta}_{t|t-1}$ no longer plays a role (can be seen by substituting the original expression for the prediction error).

[4] If (and only if) there is no measurement error then $P_{t|t} = 0$ for all $t > 1$, and $P_{t+1|t} = \Psi$.

Since Θ is a diagonal matrix with measurement error variances $\{\theta_{11}, ..., \theta_{pp}\}$ on the diagonal, $\Theta^{\frac{1}{2}}$ is a diagonal matrix with entries $\{\sqrt{\theta_{11}}, ..., \sqrt{\theta_{pp}}\}$. W_t is a diagonal matrix of weights $\{w_{1t}, ..., w_{it}, ..., w_{pt}\}$, which are calculated as

$$w_{it} = \frac{\psi_H(s_{it} - b_{it}\hat{\eta}_{t|t-1})}{s_{it} - b_{it}\hat{\eta}_{t|t-1}} \tag{17}$$

with

$$\begin{aligned} \underset{p\times 1}{s_t} &= \Theta^{-\frac{1}{2}}\, y_t \\ \underset{p\times q}{b_t} &= \Theta^{-\frac{1}{2}}\Lambda. \end{aligned}$$

Here, $\Theta^{-\frac{1}{2}}$ is again a diagonal matrix, this time with entries $\{1/\sqrt{\theta_{11}}, ..., 1/\sqrt{\theta_{pp}}\}$, s_{it} is the i-th element of vector s_t and b_{it} is the i-th row of matrix b_t. The function $\psi_H(x)$ is the Huber ψ-function that assigns function values to a vector x in an element-wise fashion such that

$$\begin{aligned} \psi_H(x) &= (\psi_H(x_1), ..., \psi_H(x_i), ..., \psi_H(x_n))^t \\ \psi_H(x_i) &= \begin{cases} x_i & \text{if} \quad |x_i| < k \\ sgn(x_i)\, k & \text{otherwise} \end{cases} \end{aligned}$$

The function $sgn(x_i)$ gives the sign of x_i (i.e., $sgn(-4) = -1$, $sgn(125) = 1$), and k is a threshold, which values larger in absolute value are set to. If k is set to some large value so that no values x_i are changed by the Huber ψ-function, then all weights will be 1, W_t will be an identity matrix, and $F_t^r = F_t$.

Note that Crevits and Croux (2019) assume $\alpha = 0$ and $\tau = 0$. The weight matrix W_t should look different if $\tau \neq 0$, but not if $\alpha \neq 0$ (the latter is already part of the predicted state). In his R function *cipraR*, Crevits includes a measurement intercept (i.e., a mean), which he substracts from the data before filtering, and adds back in afterwards.

The above weights are a function of the prediction error. For a univariate model, this can be seen directly, since

$$s_t - b_t\hat{\eta}_{t|t-1} = \frac{y_t}{\sqrt{\theta}} - \frac{\lambda\,\hat{\eta}_{t|t-1}}{\sqrt{\theta}} = \frac{y_t - \lambda\,\hat{\eta}_{t|t-1}}{\sqrt{\theta}} = \frac{v_t}{\sqrt{\theta}}. \tag{18}$$

So, what gets possibly changed by the Huber ψ-function is the prediction error relative to / scaled by the standard derivation of the measurement error. The more this term gets changed by the Huber ψ-function, the larger the discrepancy between the numerator and denominator of the weight w_t, and the smaller the weight than one (numerator is smaller than or equal to the denominator). If the weight, which appears in the denominator in Equation 16, gets smaller than one, then that effectively means that the measurement error variance gets multiplied by a factor larger than one (the reciprocal of the weight). Hence, the current (outlying) observation has less impact in updating the state estimate. The analogue holds for multivariate processes.

Comparing the classical and robust Kalman filter. The following figure contrasts the behavior of the classical and the robust KF visually.

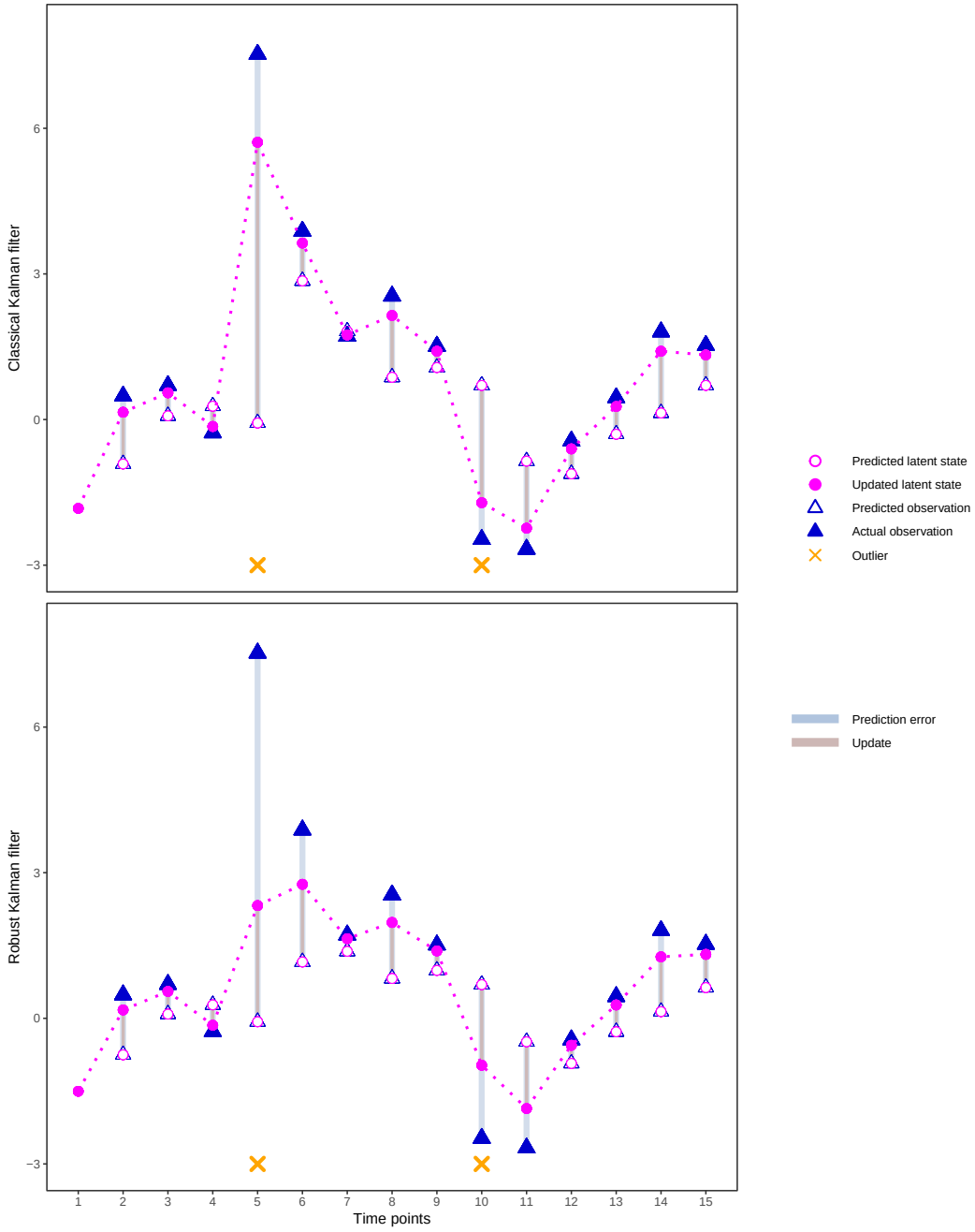


Figure 1

The classical and robust Kalman filter applied to the same time series.

1.2.3 Robust SSM implementation in R

For robust SSM on basis of the MTL estimator, we make use of the *OpenMx* option that allows users to define their own fit functions in *R* (see documentation for *OpenMx::mxFitFunctionR* here). Alternatively, the base *R* function *optim* can also be used for parameter estimation (see

documentation for *simARO::buildRobSSM*). Both fit functions use the (classical) KF and the trimmed likelihood function as implemented in the *simARO* package, *simARO::KF*, and *simARO::getMinLL*.

For the classical case (i.e., $\alpha = 0$, $k = \infty$), we can numerically check these *simARO*-functions by comparing their results to results obtained via *OpemMx*'s SSM functionality. As can be seen, the KF and log-likelihood results are exactly the same given the same data and model parameter values:

```
# simulate some data
set.seed(123)
y <- data.frame(y = rnorm(100, 5, 1))

# 'fit' (with fixed parameter values) an AR(1) model with measurement error
# using SSM in OpenMx
AR1model <- OpenMx::mxModel(
  name = "AR1model",
  mxData(observed = cbind(y = y, constant = 1), type = 'raw'),
  manifestVars = 'y',
  mxMatrix('Full', nrow = 1, ncol = 1, values = .2, free = FALSE, labels = "b11", name = 'B'),
  mxMatrix('Full', nrow = 1, ncol = 1, values = 4, free = FALSE, labels = "a1", name = 'a'),
  mxMatrix('Full', nrow = 1, ncol = 1, values = 1, free = FALSE, labels = "p11", name = 'P'),
  mxMatrix('Full', nrow = 1, ncol = 1, values = 0.1, free = FALSE, labels = "th11", name = 'Th'),
  mxMatrix("Full", nrow = 1, ncol = 1, free = FALSE, labels = "data.constant", name = "u"),
  mxMatrix('Iden', nrow = 1, ncol = 1, name = 'L', dimnames = list('y',paste0('lat','y'))),
  mxMatrix("Zero", nrow = 1, ncol = 1, name = "ta"),
  mxMatrix("Full", nrow = 1, ncol = 1, values = 5, free = FALSE, name = "x0"),
  mxMatrix("Full", nrow = 1, ncol = 1, values = 1, free = FALSE, name = "P0"),
  mxExpectationStateSpace(x0="x0", P0="P0", u="u", A="B", B="a", Q="P", C="L", D="ta", R="Th"),
  mxFitFunctionML()
)
fitAR1model <- OpenMx::mxRun(AR1model)

# get the min2ll value
min2ll_omx <- fitAR1model$output$Minus2LogLikelihood

# re-calculate the Kalman filter results for this model
resKF_omx <- OpenMx::mxKalmanScores(AR1model)

# calculate the Kalman filter results via simARO
KF_args <- list(
  Y = y,
  x0 = AR1model$x0$values,
  P0 = AR1model$P0$values,
  a = AR1model$a$values,
  B = AR1model$B$values,
  P = AR1model$P$values,
  Th = AR1model$Th$values
)
resKF_simaro <- do.call(eval(parse(text = "simARO::KF")),  KF_args)
resKFrob_simaro <- do.call(eval(parse(text = "simARO::KF")),  append(KF_args, list(rob = TRUE, k = Inf)))

# also calculate the min2ll value using simARO
min2ll_simaro <- simARO::getMinLL(
  nt = nrow(y),
  v = unlist(resKF_simaro$v),
  matF = c(unlist(resKF_simaro$matF))
  )

# compare the min2ll values
min2ll_omx
```

```
## [1] 272.0457
```

```
min2ll_simaro
```

```
## [1] 272.0457
```

```r
# compare the Kalman filter predictions and updates
rbind(resKF_omx$xPredicted |> c(), resKF_simaro$x.pre |> unlist(),
      resKFrob_simaro$x.pre |> unlist())[, sample(1:100, 5)]
```

```
##          t=29     t=93     t=88     t=15     t=87
## [1,] 5.030555 5.103084 5.200572 5.021475 5.059655
## [2,] 5.030555 5.103084 5.200572 5.021475 5.059655
## [3,] 5.030555 5.103084 5.200572 5.021475 5.059655
```

```r
rbind(resKF_omx$PPredicted |> c(), resKF_simaro$P.pre |> unlist(),
      resKFrob_simaro$P.pre |> unlist())[, sample(1:100, 5)]
```

```
##          t=53     t=74     t=47     t=19     t=66
## [1,] 1.003638 1.003638 1.003638 1.003638 1.003638
## [2,] 1.003638 1.003638 1.003638 1.003638 1.003638
## [3,] 1.003638 1.003638 1.003638 1.003638 1.003638
```

```r
rbind(resKF_omx$xUpdated |> c(), resKF_simaro$x.upd |> unlist(),
      resKFrob_simaro$x.upd |> unlist())[, sample(1:100, 5)]
```

```
##          t=92     t=35     t=51     t=21    t=48
## [1,] 5.515421 5.761875 5.229222 4.021339 4.56866
## [2,] 5.515421 5.761875 5.229222 4.021339 4.56866
## [3,] 5.515421 5.761875 5.229222 4.021339 4.56866
```

```r
rbind(resKF_omx$PUpdated |> c(), resKF_simaro$P.upd |> unlist(),
      resKFrob_simaro$P.upd |> unlist())[, sample(1:100, 5)]
```

```
##            t=41       t=58       t=83       t=10       t=54
## [1,] 0.09093905 0.09093905 0.09093905 0.09093905 0.09093905
## [2,] 0.09093905 0.09093905 0.09093905 0.09093905 0.09093905
## [3,] 0.09093905 0.09093905 0.09093905 0.09093905 0.09093905
```

Also, if the classical AR(1) model is freely estimated, we get highly similar results (parameter point estimates, standard error estimates and model fit):

```r
# fit model using SSM in OpenMx (with freely estimated parameters)
AR1model_omx <- OpenMx::omxSetParameters(AR1model, labels = c("b11", "a1", "p11"), free = TRUE)
fitAR1model_omx <- OpenMx::mxRun(AR1model_omx)
summary(fitAR1model_omx)
```

```
## Summary of AR1model
##
## free parameters:
##   name matrix row col    Estimate Std.Error A
## 1  b11      B   1   1 -0.02756296 0.1124396
## 2   a1      a   1   1  5.23094109 0.5805104
## 3  p11      P   1   1  0.72435741 0.1165901
##
## Model Statistics:
##                |  Parameters  |  Degrees of Freedom  |  Fit (-2lnL units)
##        Model:              3                    97              264.4784
##    Saturated:              2                    98                    NA
## Independence:              2                    98                    NA
## Number of observations/statistics: 100/100
##
## Information Criteria:
##       |  df Penalty  |  Parameters Penalty  |  Sample-Size Adjusted
## AIC:        70.4784              270.4784                270.7284
## BIC:      -182.2231              278.2939                268.8192
## CFI: NA
## TLI: 1   (also known as NNFI)
## RMSEA:  0  [95% CI (NA, NA)]
## Prob(RMSEA <= 0.05): NA
```

```
## To get additional fit indices, see help(mxRefModels)
## timestamp: 2024-12-13 22:42:28
## Wall clock time: 0.07050085 secs
## optimizer:  SLSQP
## OpenMx version number: 2.21.1
## Need help?  See help(mxSummary)
```

```
# fit model using simARO functionality
AR1model_simaro <- buildRobSSM(
  data = y,
  Yname = 'y',
  ne = 1,
  ny = 1,
  B.init = AR1model_omx$B$values,
  a.init = AR1model_omx$a$values,
  P.init = AR1model_omx$P$values,
  Th.init = AR1model_omx$Th$values,
  Th.est = AR1model_omx$Th$free,
  x0 = AR1model_omx$x0$values,
  P0 = AR1model_omx$P0$values,
  k = Inf,
  trim.useOrgLL = FALSE
  )

fitAR1model_simaro <- OpenMx::mxRun(AR1model_simaro)
summary(fitAR1model_simaro)
```

```
## Summary of untitled1
##
## free parameters:
##   name matrix row col    Estimate Std.Error A  lbound ubound
## 1  b11      B   1   1 -0.02756299 0.1124414
## 2   a1      a   1   1  5.23094162 0.5805193
## 3  p11      P   1   1  0.72435773 0.1165902   0.00001 100000
##
## Model Statistics:
##                |  Parameters  |  Degrees of Freedom  |  Fit (-2lnL units)
##        Model:              3                     -3              264.4784
##    Saturated:             NA                     NA                    NA
## Independence:             NA                     NA                    NA
## Number of observations/statistics: 100/0
##
## Information Criteria:
##       |  df Penalty  |  Parameters Penalty  |  Sample-Size Adjusted
## AIC:       270.4784               270.4784                270.7284
## BIC:       278.2939               278.2939                268.8192
## CFI: NA
## TLI: 1   (also known as NNFI)
## RMSEA:  0  [95% CI (NA, NA)]
## Prob(RMSEA <= 0.05): NA
## To get additional fit indices, see help(mxRefModels)
## timestamp: 2024-12-13 22:42:33
## Wall clock time: 4.903164 secs
## optimizer:  SLSQP
## OpenMx version number: 2.21.1
## Need help?  See help(mxSummary)
```

For the robust case, we have numerically checked our implementation using the author's original code.

2 Supplementary materials to the simulation study

Here we provide additional information on the simulation study. This concerns descriptions of those simulation settings that were only briefly listed in the manuscript, but also more details on some of the already described settings. We also present additional results, some of which are referred to in the manuscript.

2.1 Design

2.1.1 Data generation

Sample sizes. Next to a sample size of $T = 80$, we also included sample sizes of $T = 160$ and $T = 800$ to check the generalizability of results to larger samples.

Mixed outliers. In addition to having either innovation or measurement outliers, we also included mixed innovation and measurement outliers. To generate those, the two outlier indicator variables were designed to both take effect within a replication. The marginal probabilities to generate the outlier indicator variables were of the same value as in the case of single-type outliers (i.e., $\{0.1, 0.2, 0.3\}$), but here they concern the probabilities of at least one of the two indicators being one. The overlap of the two outlier indicators was always set to 0.5.

As an example, consider a marginal probability of 0.2: If $T = 160$ then we have exactly 32 contaminated occasions, of which 8 are contaminated with only measurement outliers, 8 with only innovation outliers, and 16 are contaminated with both simultaneously.

Temporally dependent outlier-indicators. For the conditions involving single outlier types we, next to randomly occurring ones, include outliers that occur with medium and high temporal dependence. To generate outlier-indicator variables with different degrees of temporal dependence, we vary the conditional probability of $x_{\cdot_t}$ being one given that $x_{\cdot_{t-1}=1}$ denoted as $P_{x_{\cdot_t}=1|x_{\cdot_{t-1}}=1}$. Whereas this equals the marginal probability of $x_{\cdot_t} = 1$ for randomly occurring outliers, hence $P_{x_{\cdot_t}=1|x_{\cdot_{t-1}}=1} = P_{x_{\cdot_t}=1}$, we set $P_{x_{\cdot_t}=1|x_{\cdot_{t-1}}=1} = \{P_{x_{\cdot_t}=1} + 0.2, P_{x_{\cdot_t}=1} + 0.4\}$ to obtain outlier-indicator series with medium and high temporal dependence (Bodner, Tuerlinckx, Bosmans, & Ceulemans, 2021).

Since we use a markov chain and hence a stochastic process to generate these temporally dependent outlier indicator data, we do not have control over the marginal or the conditional probabilities being exact for any given finite time series generated. For the marginal probabilities, we restored exact probabilities by over-generating outlier-indicator series and retaining only those with the correct marginal frequencies. This selection procedure should however not have systematic influences on the conditional probabilities, for which we found no way to implement them in an exact manner. This means that the actual temporal dependencies vary randomly across simulation replications and averages per design cell show random deviations from the nominal values (i.e., sampling error). Simulation results will reflect this additional variation.

No outliers at the first occasion. Under all settings, we prevented that outliers could occur at the first time point. The reason for this is that the first observation is handled differently during model fitting, as described in the following section.

2.1.2 Models fitted

Parameter starting values. All models implemented in *OpenMx* (i.e., the true, classical, cML and MTL AR(1) models) are estimated in an iterative fashion via one of the package's numerical optimizers (i.e., we pick *SLSQP*). A numerical optimization procedure requires setting starting values for the model parameters. Per simulated data set, we draw these randomly from uniform distributions such that the starting values for all common parameters are the same across all *OpenMx* models. To obtain good starting values, we set the uniform distribution bounds so that a range of admissible and plausible values are spanned per parameter. In addition we make use of *OpenMx*'s own functionality

to converge on good parameter starting values implemented in the function *mxTryHard*. Under the MM AR(1) model parameter starting values are obtained internally via initial robust estimates (Maechler et al., 2021). These are thus not the same as for the *OpenMx* models.

Handling of initial observations. For the SSM and thus again *OpenMx* models the first observation is included during estimation and handled conditional on the initial values that one needs to provide for the KF (e.g., Chow, Ho, Hamaker, & Dolan, 2010; Hardt, Hecht, Oud, & Voelkle, 2019; Hunter, 2018). Here we set these initial values, the state estimate for $t = 0$ and the associated prediction error variance, to the observed mean of each simulated data set and a value of 10, respectively, yielding a rather uninformative initialization (Chow et al., 2010). The MM AR(1) model on the contrary cannot handle missing predictor values (i.e., $y_{t-1}|t = 1$) and thus (per default) automatically discards the first row of the data. These differences in how initial observations are handled might have slight effects on the results under smaller sample size settinge and hence possibly the $T = 80$ condition. But at the same time such differences also just reflect how the models would be applied to actual data.

Implementation of robust SSM. The implementation of robust SSM in *R* is described in the previous section 1.2.3.

2.1.3 Assessing model performance

Mean-squared prediction error. Next to estimation accuracy in terms of the mean squared estimation error (MSE) of the AR effect, we also assess predictive accuracy of models with respect to unseen data (Yarkoni & Westfall, 2017). So, for each simulation replication and design cell we actually generate two independent data sets, one to which the model is fitted and one that we calculate the predictive accuracy of the fitted model for. For the generation of the latter prediction data we use the same baseline population models and according parameter values as for the estimation data, and generate a time series of corresponding length. Using the baseline models for data generation means that outliers are absent from the prediction data under all conditions.

We measure predictive accuracy using the mean squared prediction error (MSPE) as observed across simulation replications. The observed MSPE of any AR(1) model can readily be calculated via the Kalman filter (KF). To this end, the KF is set up based on the estimated parameter values $\hat{\omega}_{\bullet_{obs,\,i}} = \{\hat{\alpha}_{\bullet_{obs,\,i}}, \hat{\beta}_{\bullet_{obs,\,i}}, \hat{\psi}_{\bullet_{obs,\,i}}, \hat{\theta}_{\bullet_{obs,\,i}}\}$, where the bullet is a place-holder for the different possible estimators (e.g., ML, MTL, MM), and applied to the prediction data. As detailed in the manuscript, the KF then generates model-based predictions $\widetilde{y}_t|\,\hat{\omega}_{\bullet_{obs,\,i}}$ for each time point, which can simply be squared and summed to yield the observed MSPE:

$$MSPE\left(\hat{\omega}_{\bullet_{obs,\,i}}\right)_{obs,\,i} = \frac{1}{T}\sum_{t=1}^{T}\left(y_t - \widetilde{y}_t|\,\hat{\omega}_{\bullet_{obs,\,i}}\right)^2.$$

Since the KF is already used during model estimation and we have an *R* implementation at hand, it is straightforward to also use it for prediction. To initialize the KF we use the observed means of each simulated data set and a prediction error variance of 10, as we did for model fitting.

The above MSPE is calculated for each simulation replication and the mean of the individual MSPEs across all N replications is eventually used as the performance criterion. To quantify the error in our simulation due to 'sampling' random replications we bootstrap the sampling distribution of the mean observed MSPE with 500 bootstrap replicates. We report the 16^{th} and 84^{th} percentiles of the sampling distribution as two thirds of the bootstrapped MSPEs lie between those.

Outlier handling. As a third performance criterion, we inspect how the different methods handle outliers. That is, every simulated data point can be categorized as either being a vertical outlier, or a bad leverage point, or a vertical outlier and a bad leverage point, or being of another type (e.g., uncontaminated observation or a good leverage point). Per category we count the number

of data points excluded by each robust model during estimation relative to the overall number of data points (i.e., relative to the time series length T). These relative frequencies can then be compared across methods to explore sources of possible estimation (and subsequently prediction) error.

Doing the required calculations is straightforward for the cML and MTL estimators, since these actually exclude data points during estimation (i.e., they rely on discrete weighting functions). For the MM estimator, we treat data points as 'excluded' if they get an estimation weight smaller than 0.1. We expect that this rather liberal threshold allows us to get a better idea of how different data point types are handled by the MM estimator. But one should keep in mind that the displayed data points are still included during estimation, with weights *ranging* from 0 to 0.1.

Note that one could in principle extend the investigation of outlier handling by creating different categories involving good leverage points. We did not do this for now as good leverage points are harder to define than vertical outliers or bad leverage points. The reason for this is that their leveraging impact depends gradually on the population-level AR effect size and on the time passed since a vertical outlier happened.

2.2 Results

2.2.1 Non-converged modeling solutions

The figures in this section display the number of non-converged modeling solutions across all simulation conditions. Figure 2 thereby pertains to modeling solutions excluding measurement error, data with no measurement error and short time series ($T = 80$), Figure 3 to modeling solutions excluding measurement error, data with measurement error and short time series, Figure 4 to modeling solutions including measurement error, data with no measurement error and short time series, Figure 5 to modeling solutions including measurement error, data with measurement error and short time series, Figure 6 to modeling solutions excluding measurement error, data with no measurement error and medium-length time series ($T = 160$), Figure 7 to modeling solutions excluding measurement error, data with measurement error and medium-length time series, Figure 8 to modeling solutions including measurement error, data with no measurement error and medium-length time series, Figure 9 to modeling solutions including measurement error, data with measurement error and medium-length time series, Figure 10 to modeling solutions excluding measurement error, data with no measurement error and long time series ($T = 800$), Figure 11 to modeling solutions excluding measurement error, data with measurement error and long time series, Figure 12 to modeling solutions including measurement error, data with no measurement error and long time series, and Figure 13 to modeling solutions including measurement error, data with measurement error and long time series.

In all figures the number of non-converged modeling solutions (y-axis) is shown as a function of the modeling approach used (x-axis; labels including the term "noise" refer to modeling approaches accounting for measurement error). Per figure outlier types are distinguished by panel. Within each panel outlier proportions are distinguished by rows and population-level AR effect sizes by columns. Bars are colored based on the estimation method, and combined with lines of different type to indicate whether outliers occurred randomly, with medium or high temporal dependence.

Across all design cells the mean number of non-converged solutions is 2.21, the median is 0, and the maximum is 25. As noted in the manuscript, an accumulation of cases seems to happen where MTL estimation is used with too small or too large trimming proportions. Also the temporal dependence with which outliers occur seems to matter. As noted in the manuscript, an accumulation of cases seems to happen where MTL estimation is used with too small or too large trimming proportions. Also the temporal dependence with which outliers occur seems to matter.

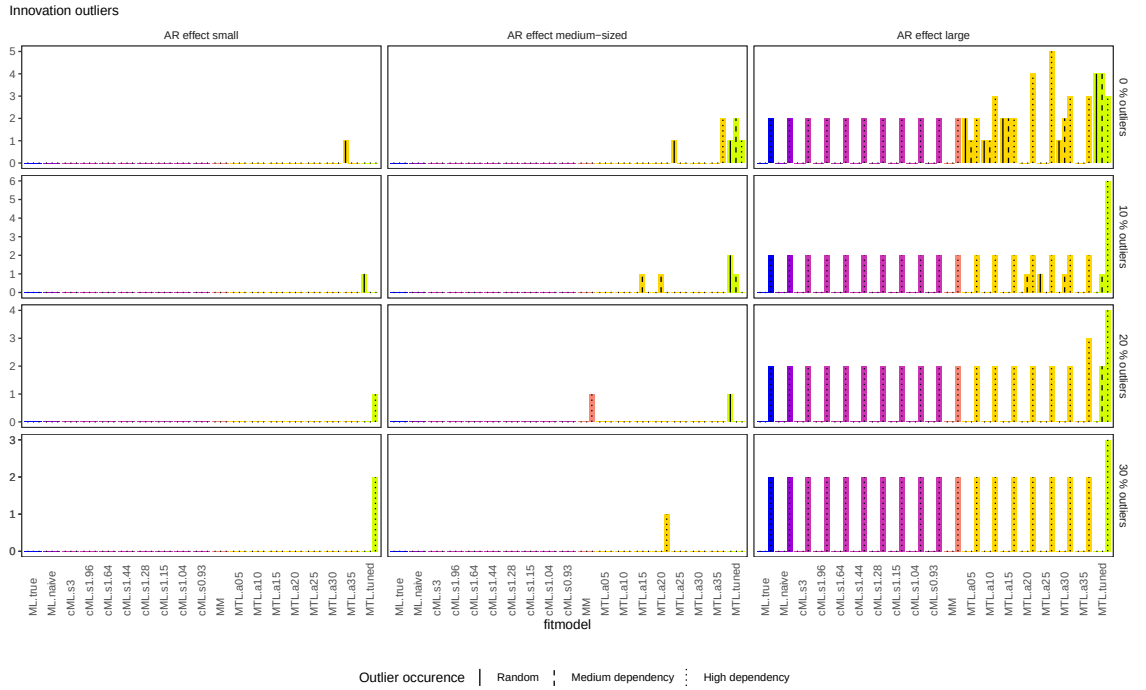
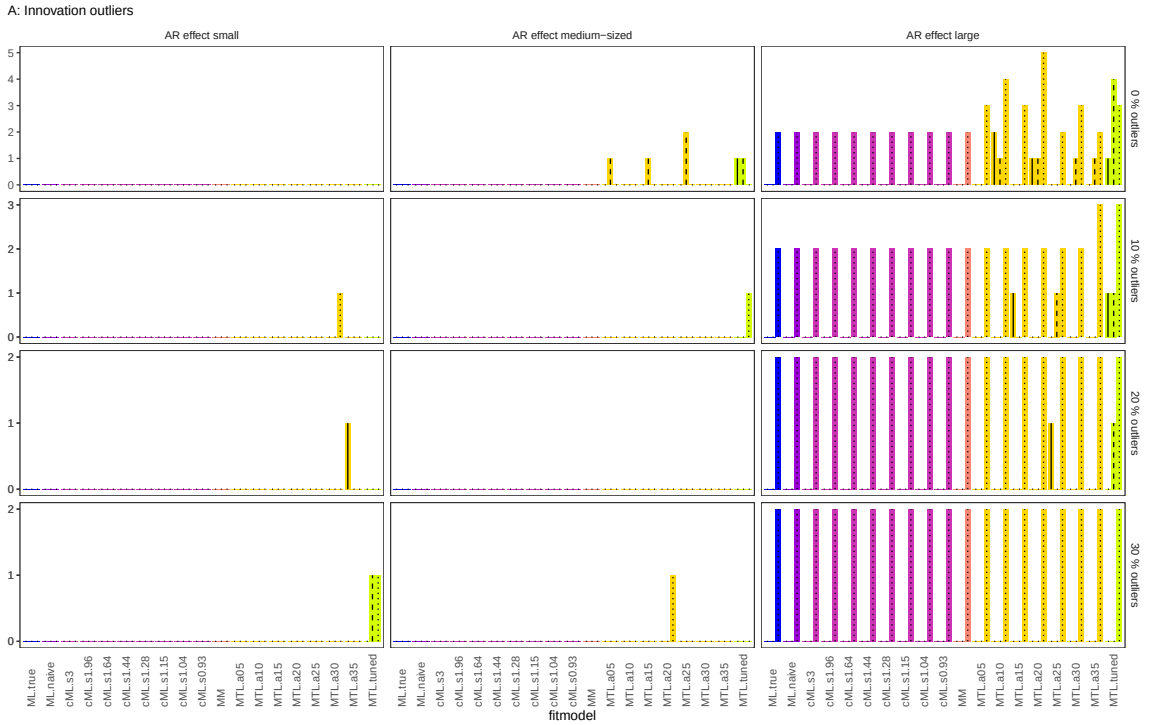


Figure 2

Number of non-converging modeling solutions excluding measurement error for data with no measurement error and short time series ($T = 80$).



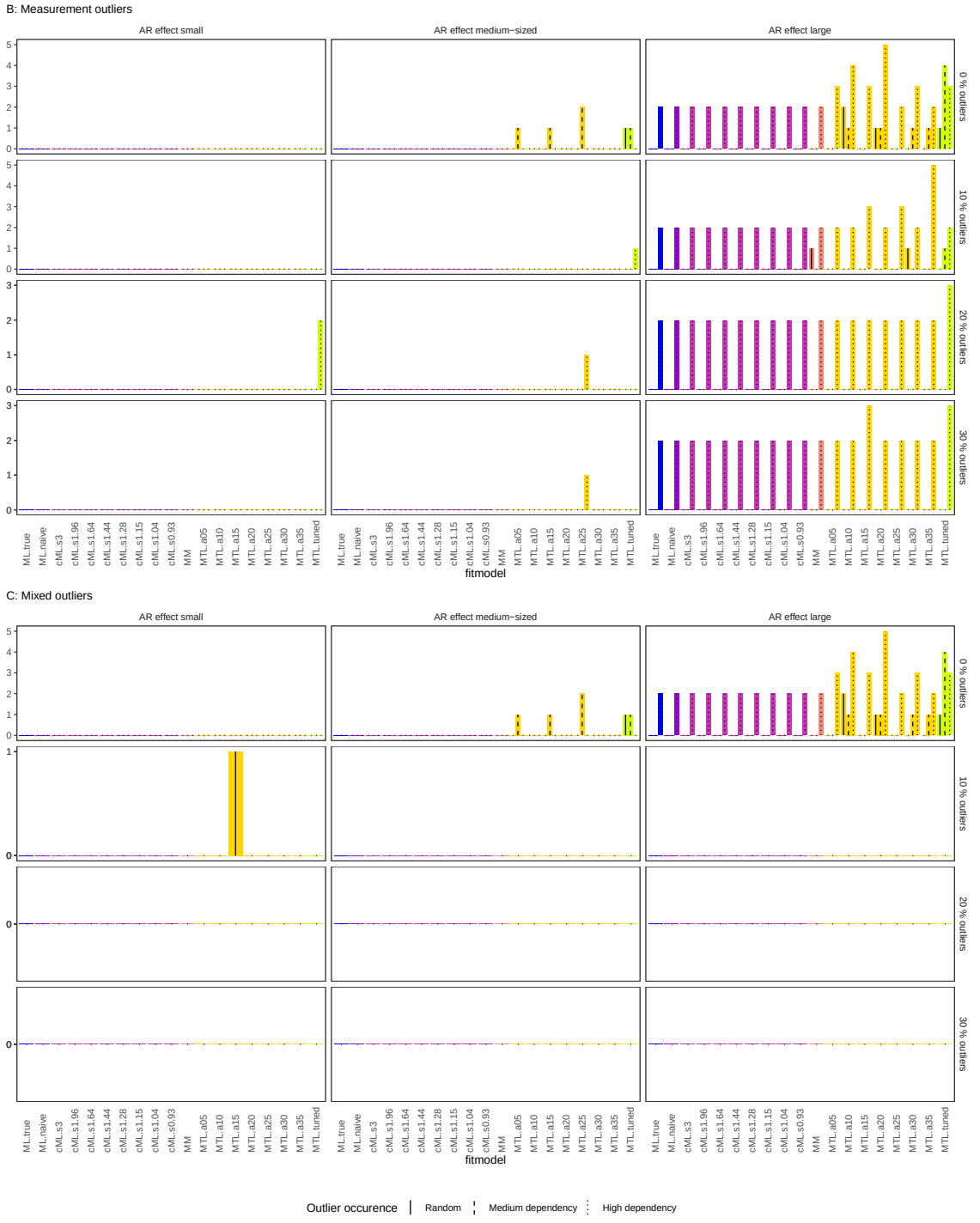


Figure 3

Number of non-converging modeling solutions excluding measurement error for data with little measurement error and short time series ($T = 80$).

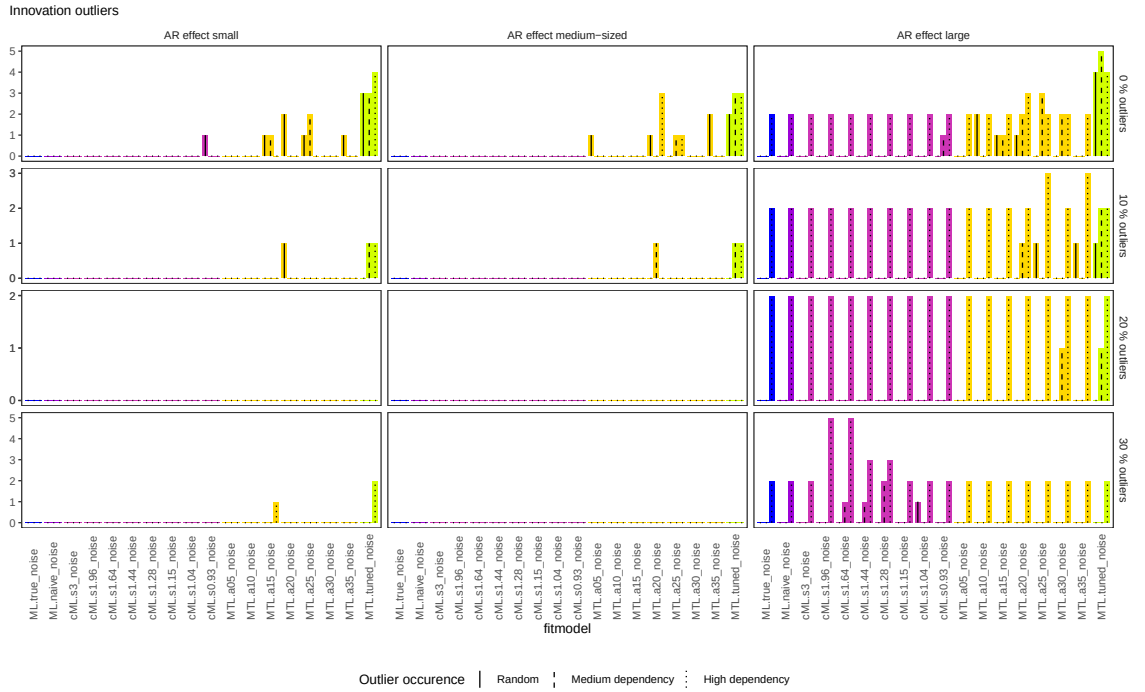


Figure 4

Number of non-converging modeling solutions including measurement error for data with no measurement error and short time series ($T = 80$).

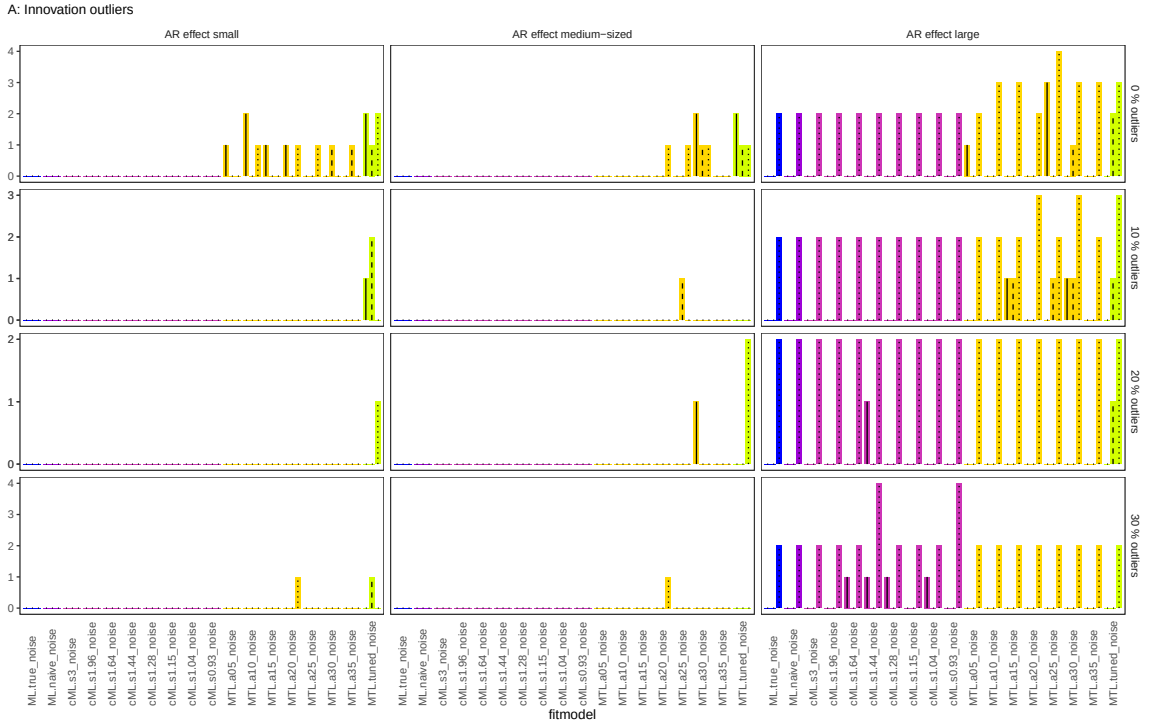


Figure 5

Number of non-converging modeling solutions including measurement error for data with little measurement error and short time series ($T = 80$).

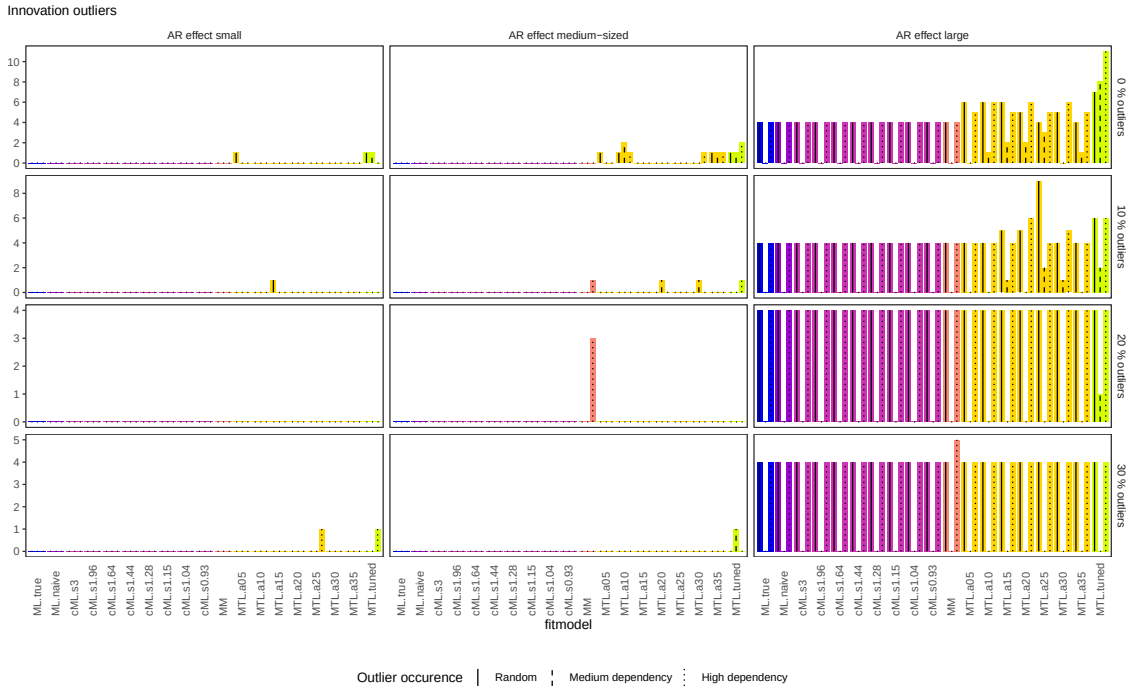
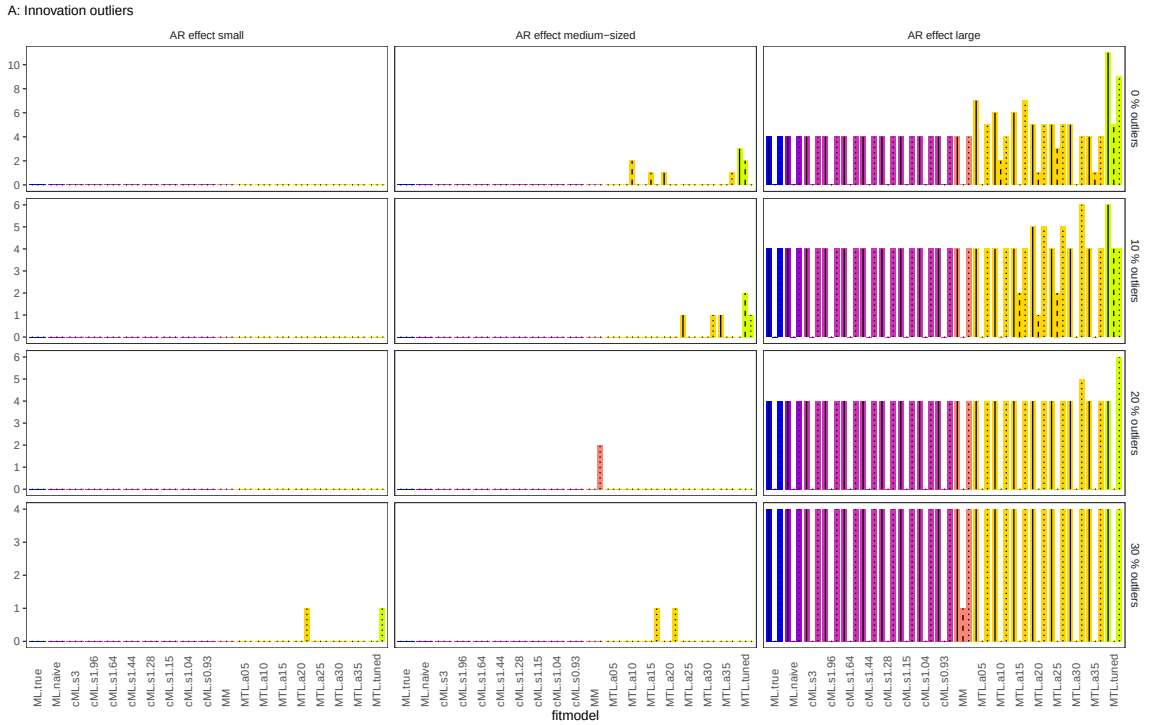


Figure 6

Number of non-converging modeling solutions excluding measurement error for data with no measurement error and medium-length time series ($T = 160$).



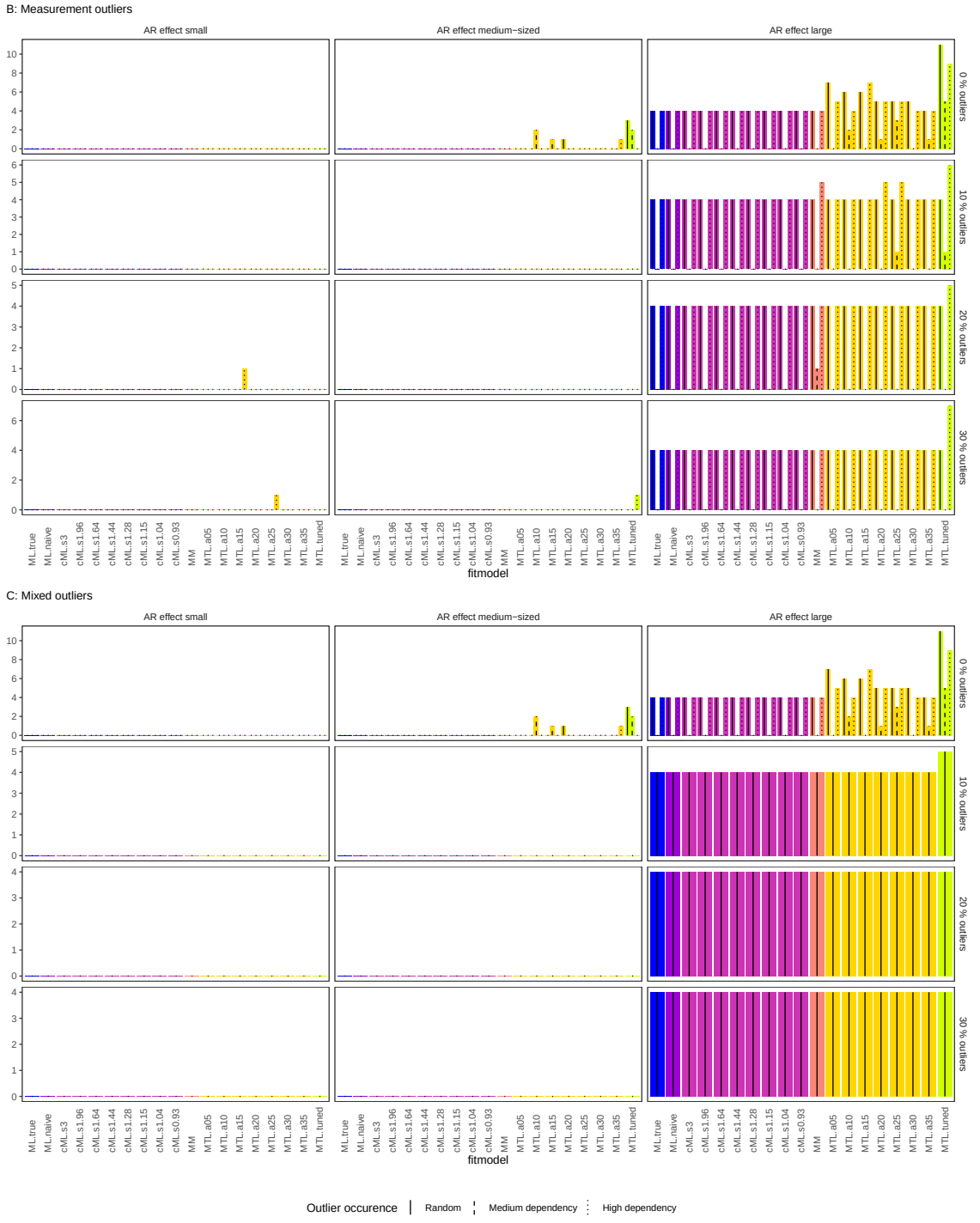


Figure 7

Number of non-converging modeling solutions excluding measurement error for data with little measurement error and medium-length time series ($T = 160$).

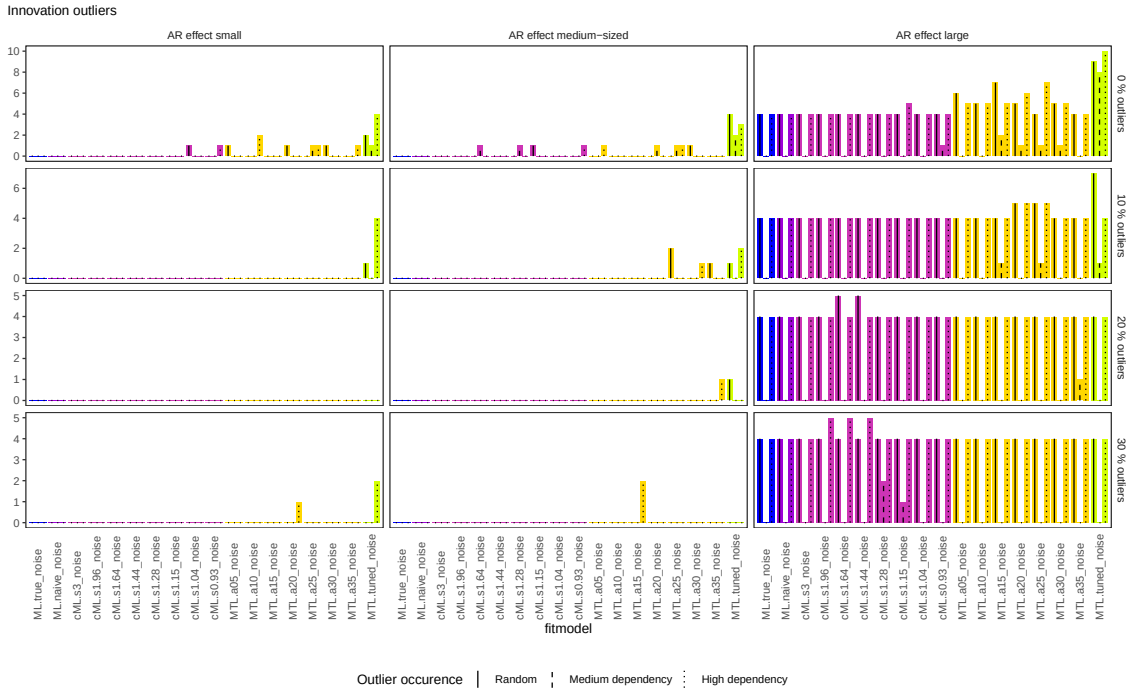
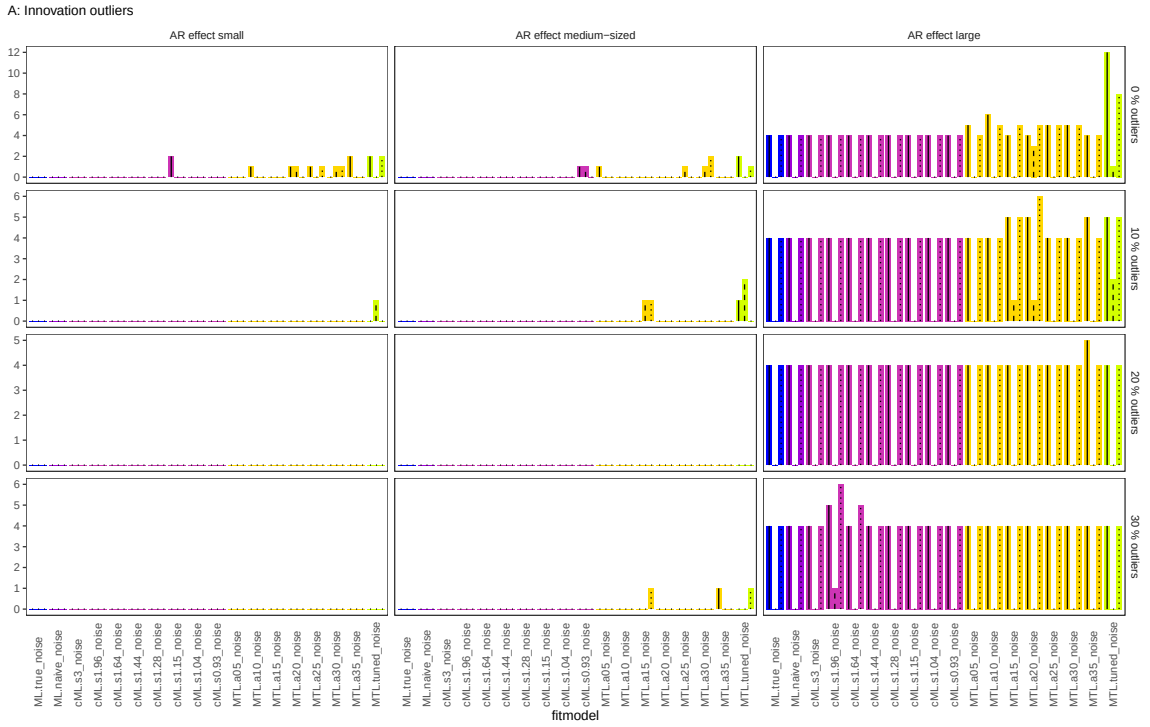


Figure 8

Number of non-converging modeling solutions including measurement error for data with no measurement error and medium-length time series ($T = 160$).



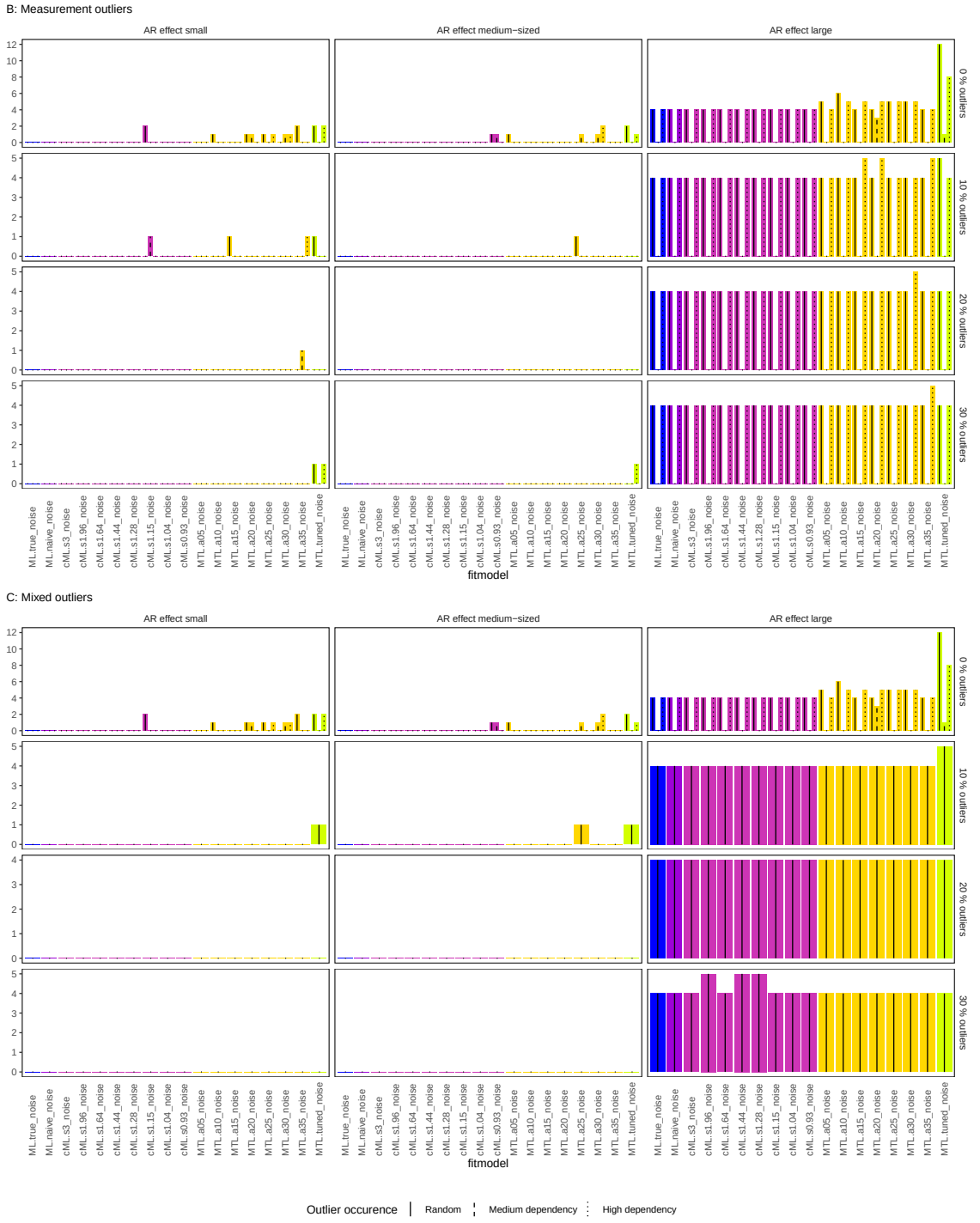


Figure 9

Number of non-converging modeling solutions including measurement error for data with little measurement error and medium-length time series ($T = 160$).

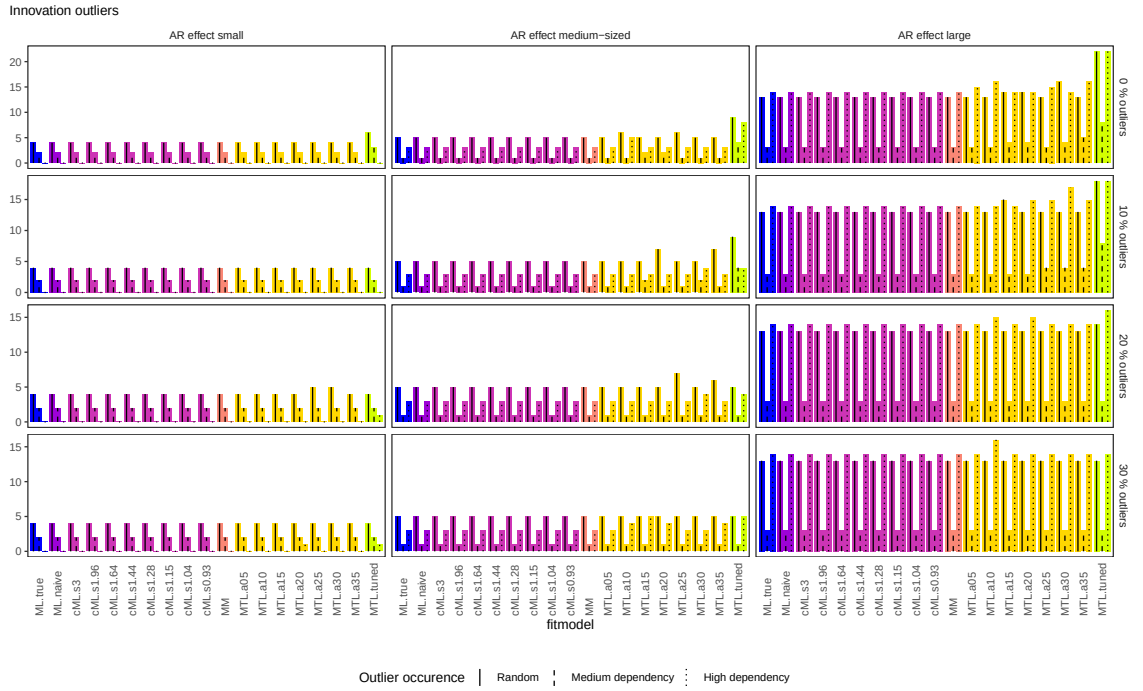
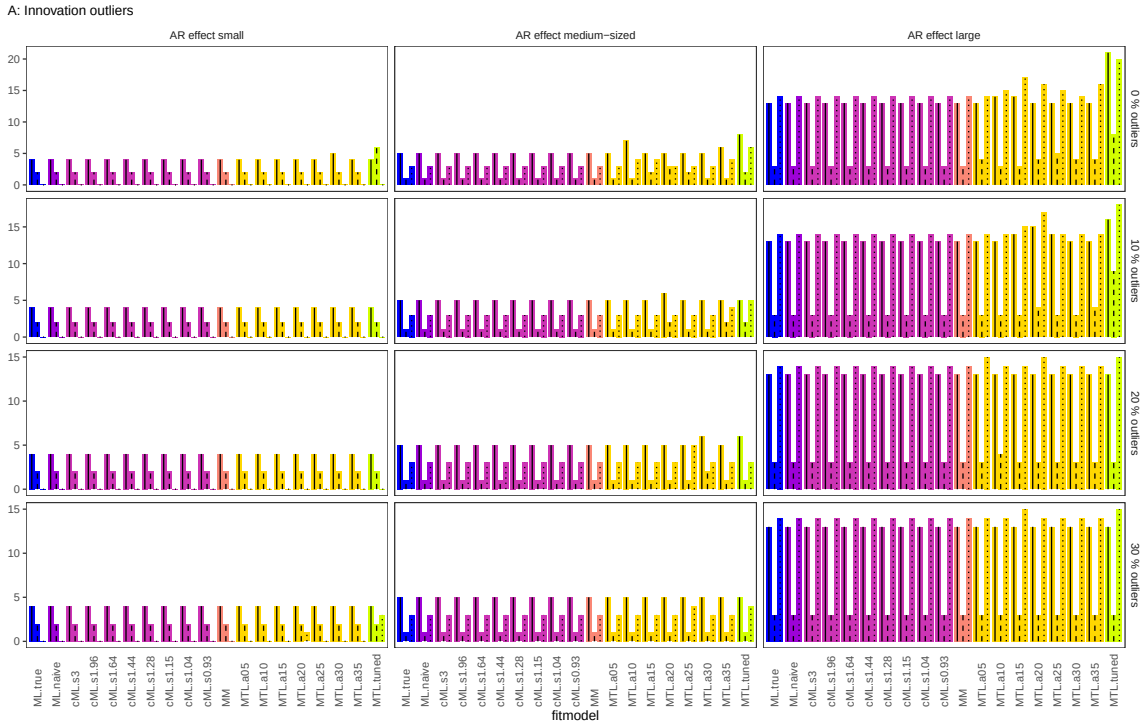


Figure 10

Number of non-converging modeling solutions excluding measurement error for data with no measurement error and long time series ($T = 800$).



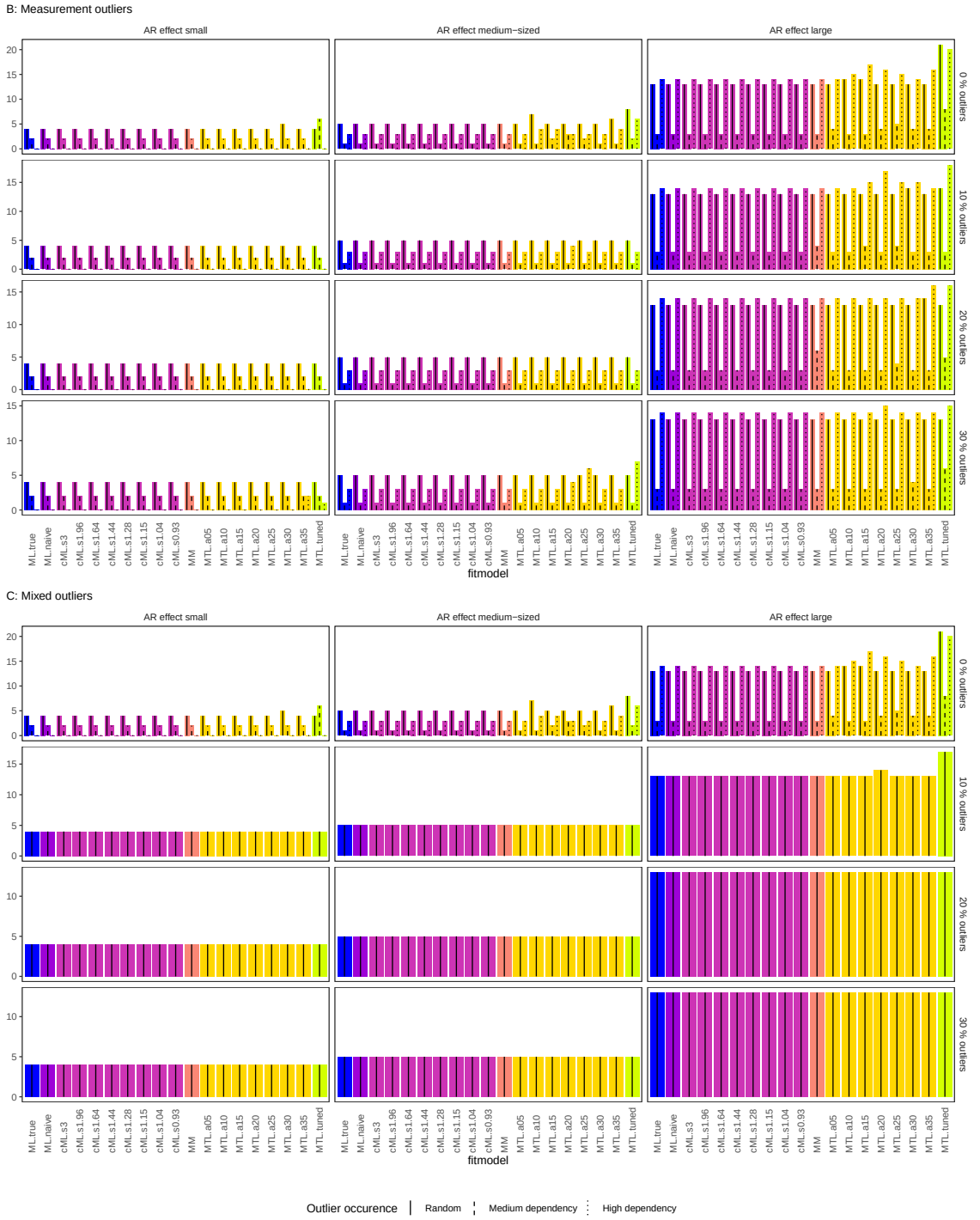


Figure 11

Number of non-converging modeling solutions excluding measurement error for data with little measurement error and long time series ($T = 800$).

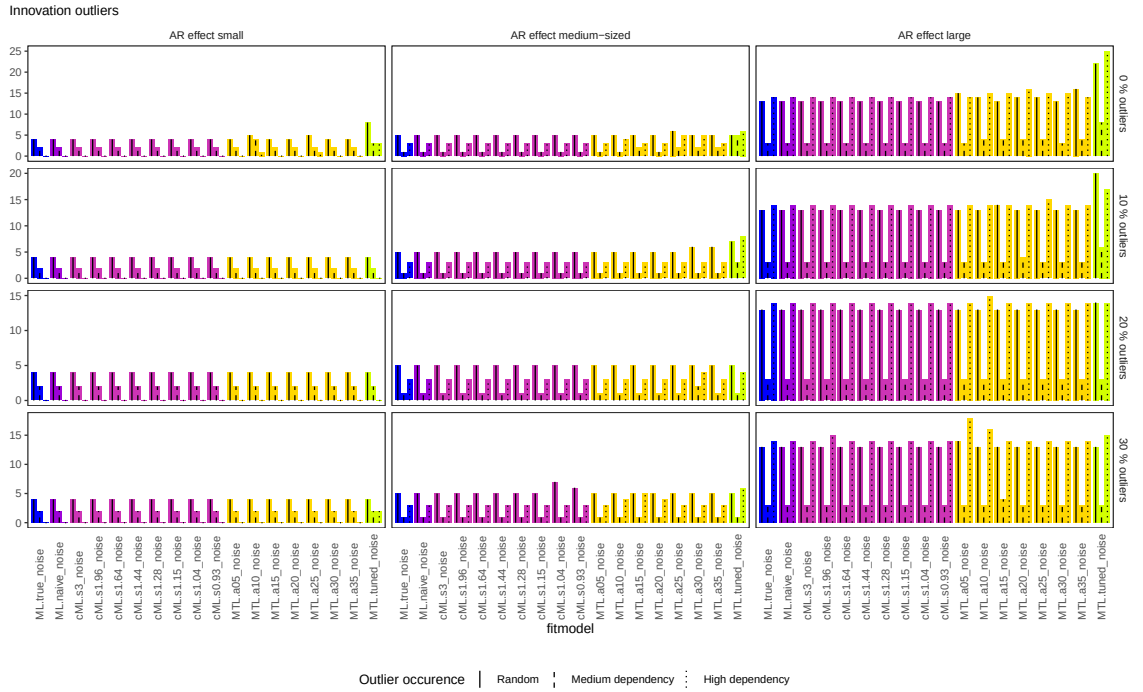
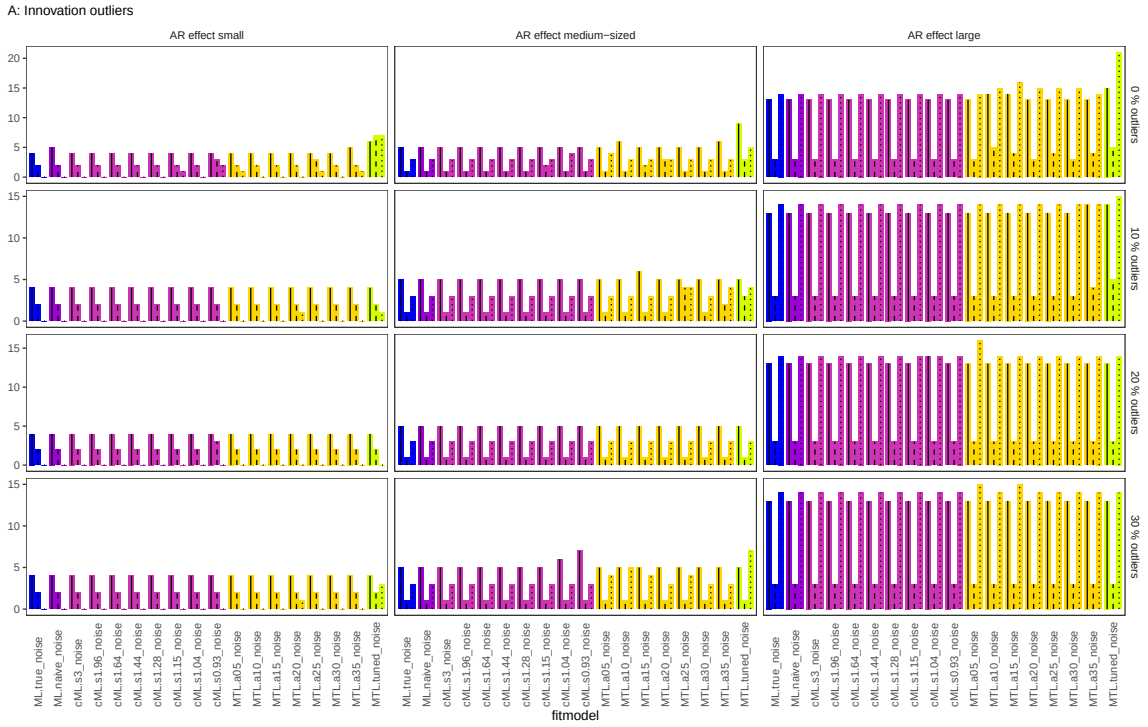


Figure 12

Number of non-converging modeling solutions including measurement error for data with no measurement error and long time series ($T = 800$).



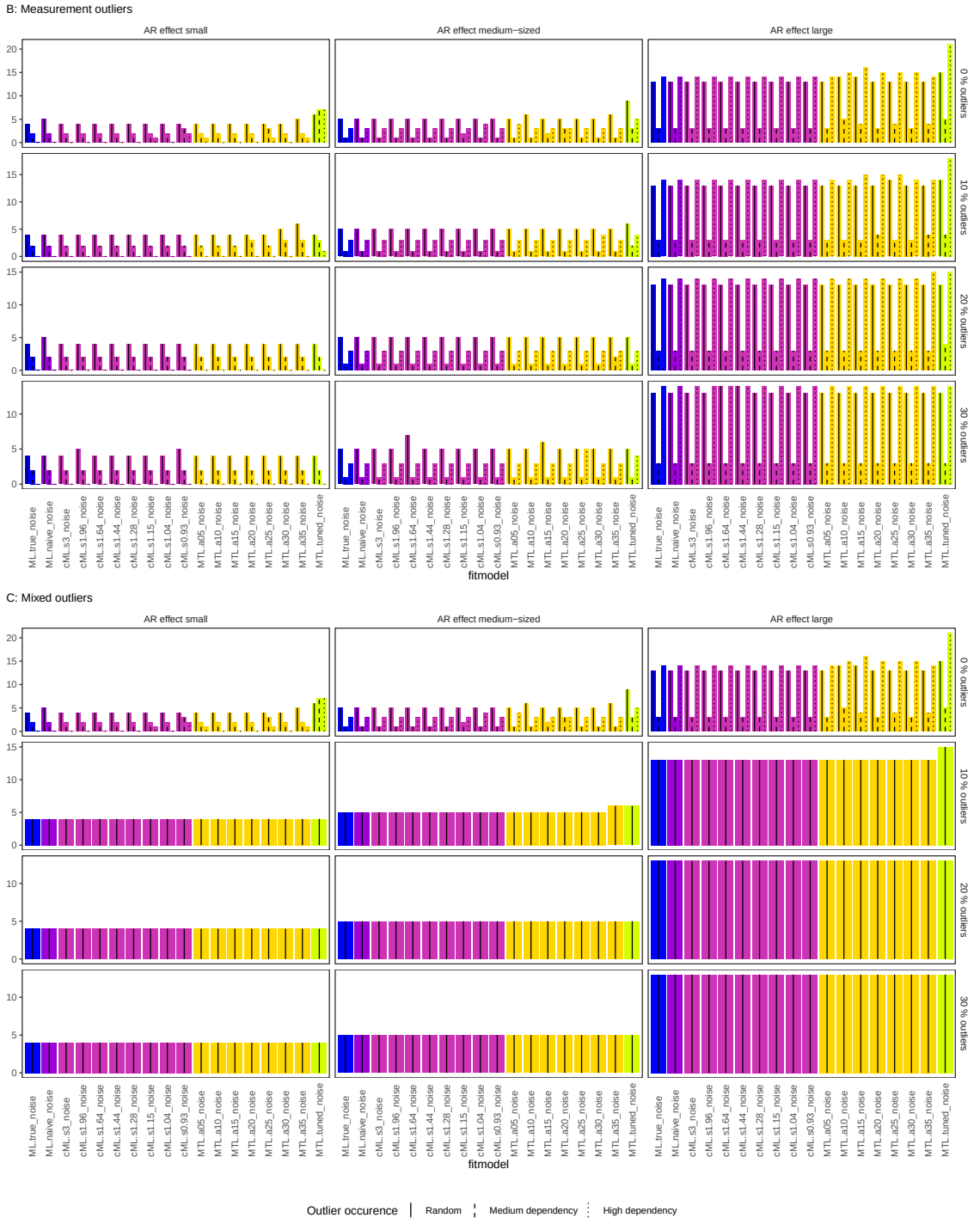


Figure 13

Number of non-converging modeling solutions including measurement error for data with little measurement error and long time series ($T = 800$).

2.2.2 Improved estimation due to innovation outliers

The figures in this section display the effects of innovation outliers on model performance in terms of the estimation accuracy of the AR parameter (MSE; Figure 14) and predictive accuracy of the overall model (MSPE; Figure 15). We limit the presentation here to short time series ($T = 80$) with randomly occurring outliers, and to models fitted without measurement error. Results on model performance given longer time series, temporally dependent outliers and modeling measurement error are shown in the following sections.

In each figure - as in the manuscript - the performance measure (y-axis) is shown as a function of the proportion of innovation outliers (x-axis). Panels A display results for measurement error-free data, panels B for measurement error-affected data. Estimation methods are distinguished by rows, population-level AR effect sizes by columns. The bars marking estimation errors are decomposed into variance (black; lower bar part) and squared bias (color depending on method; upper bar part). Dashed and solid grey lines (re-)mark the performance of the classical and the true model respectively. Unlike in the manuscript, innovation outliers are contrasted with the other outlier types (measurement and mixed) when data are affected by measurement error. Also, turquoise error bars are added visualizing the bootstrapped MSE and MSPE percentiles.

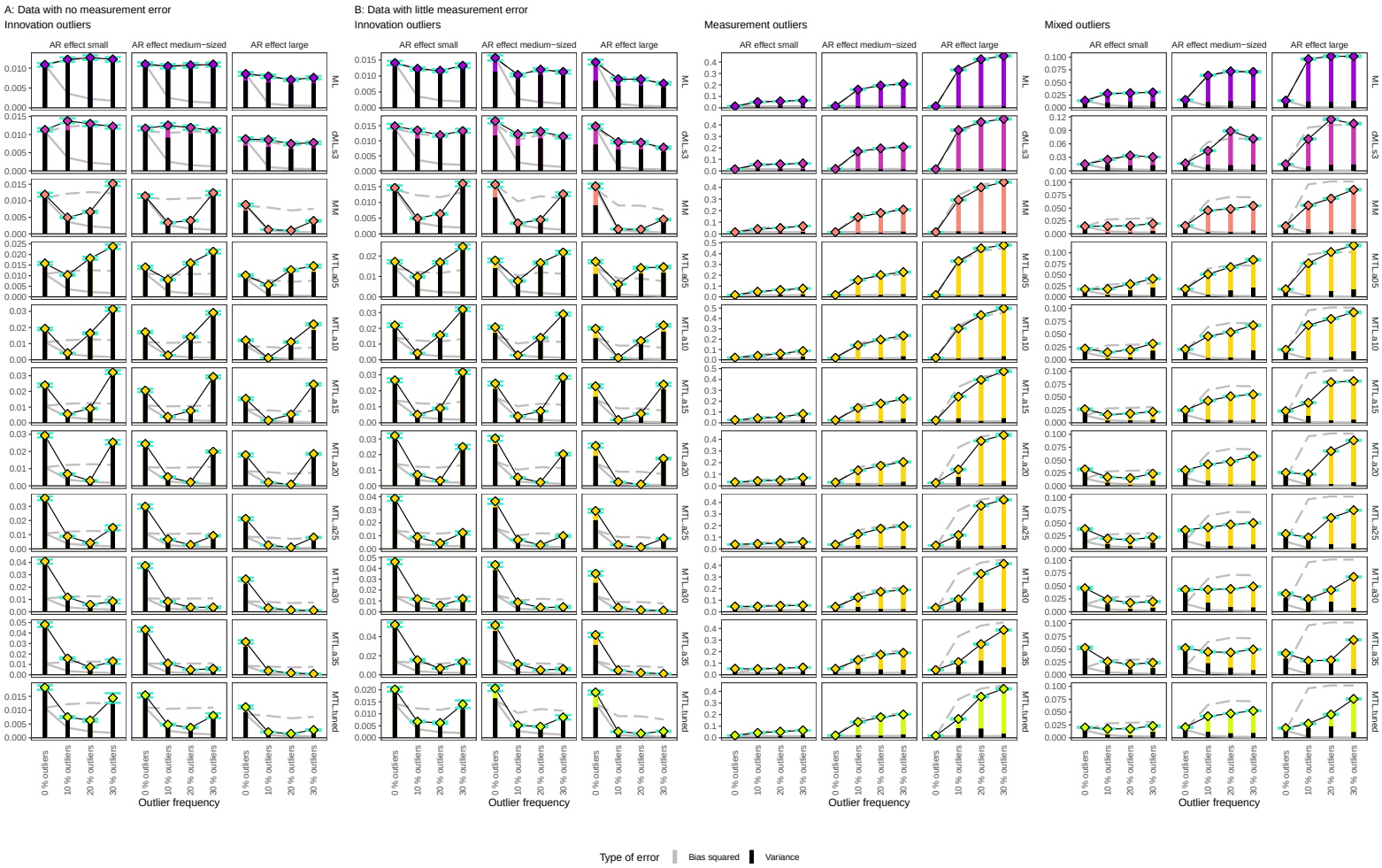


Figure 14

Estimation error AR effect for short time series ($T = 80$) with randomly occurring outliers, and models fitted without measurement error.

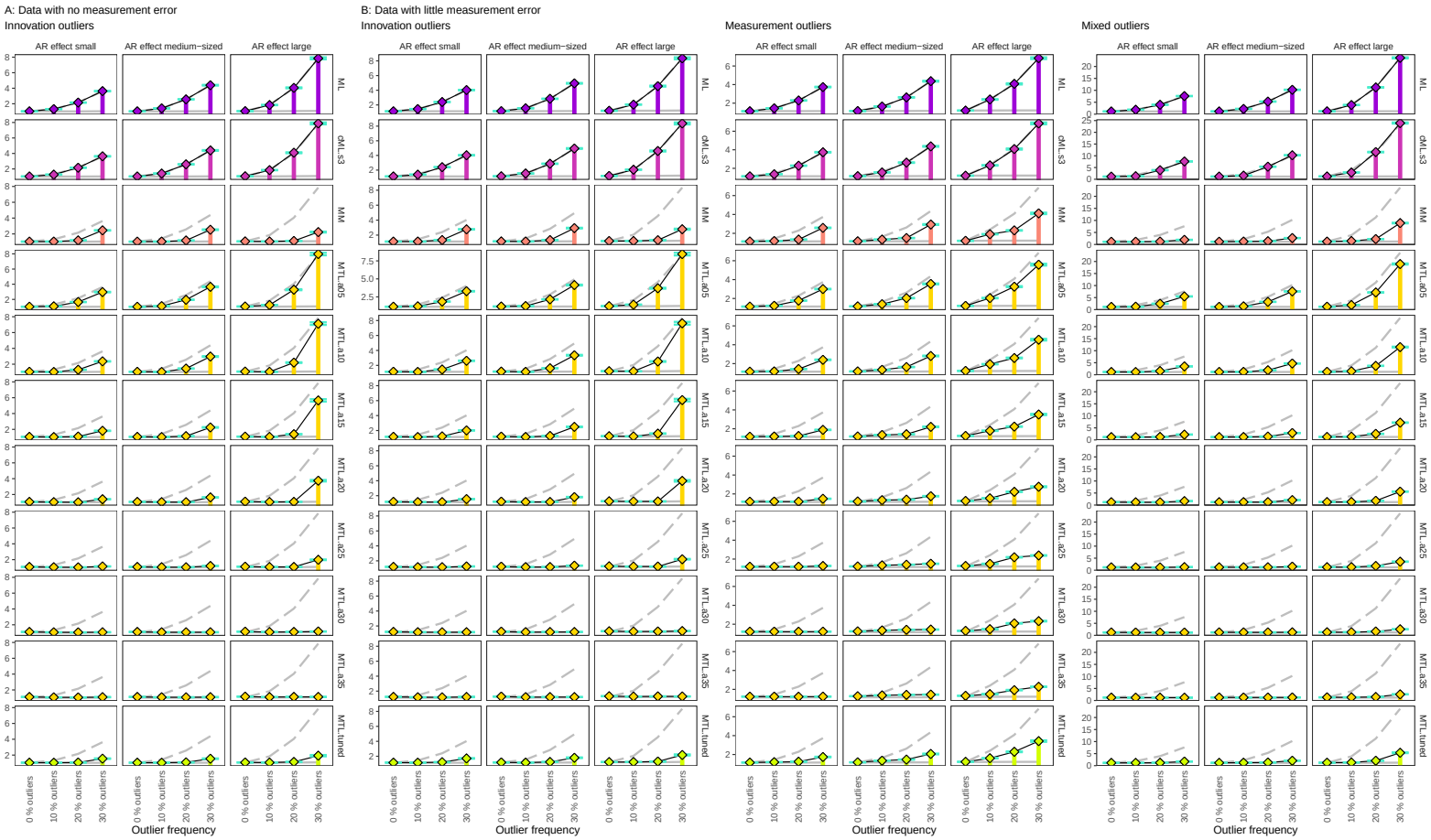


Figure 15

Prediction error for short time series ($T = 80$) with randomly occurring outliers, and models fitted without measurement error.

2.2.3 Improved estimation and prediction due to robust methods

The figures in this section display again performance of the different robust methods under different types of outliers. Here we cast model performance again in terms of the MSE of the AR parameter, the MSPE of the overall model, but also in terms of how a method handles outliers during estimation. Figures 16, 19 and 22 show the estimation error of the AR effect for short, medium-length and long time series, respectively. Figures 17, 20 and 23 show the prediction error, and Figures 18, 21 and 24 the outlier handling for short, medium-length and long time series, respectively. We limit the presentation to randomly occurring outliers, and to models fitted without measurement error. Results on model performance given temporally dependent outliers and modeling measurement error are shown in the following sections.

For the MSE- and MSPE-figures, we again show the performance measure (y-axis) as a function of the (robust) modeling approach used (x-axis). Per figure panels A display results for measurement error-free data and innovation outliers, panels B for measurement error-affected data and innovation, measurement and mixed outliers. Outliers proportions are distinguished by rows, population-level AR effect sizes by columns. The bars marking estimation errors are decomposed into variance (black; lower bar part) and squared bias (color depending on method; upper bar part). Dashed and solid grey lines (re-)mark the performance of the classical and the true model respectively. Turquoise error bars visualize the bootstrapped MSE and MSPE percentiles. Turquoise vertical lines indicate the 'winning' model with the smallest observed MSE or MSPE per tile (solid line) and those models that closely follow-up (i.e., models of which the observed MSE or MSPE is smaller than the bootstrapped 84^{th} MSE percentile of the winning model; dashed lines).

The outlier handling figures show the relative frequencies of vertical outliers (VO), bad leverage points (BP), simultaneous vertical outlier and bad leverage points (VO+BLP) and other data points (y-axis) excluded by each robust method (x-axis). The actual distribution of data points is included as a reference as the left-most bar (with axis-label "In data"). Otherwise these figures are structured similarly to the MSE- and MSPE-ones (in terms of what is shown in panels, columns and rows).

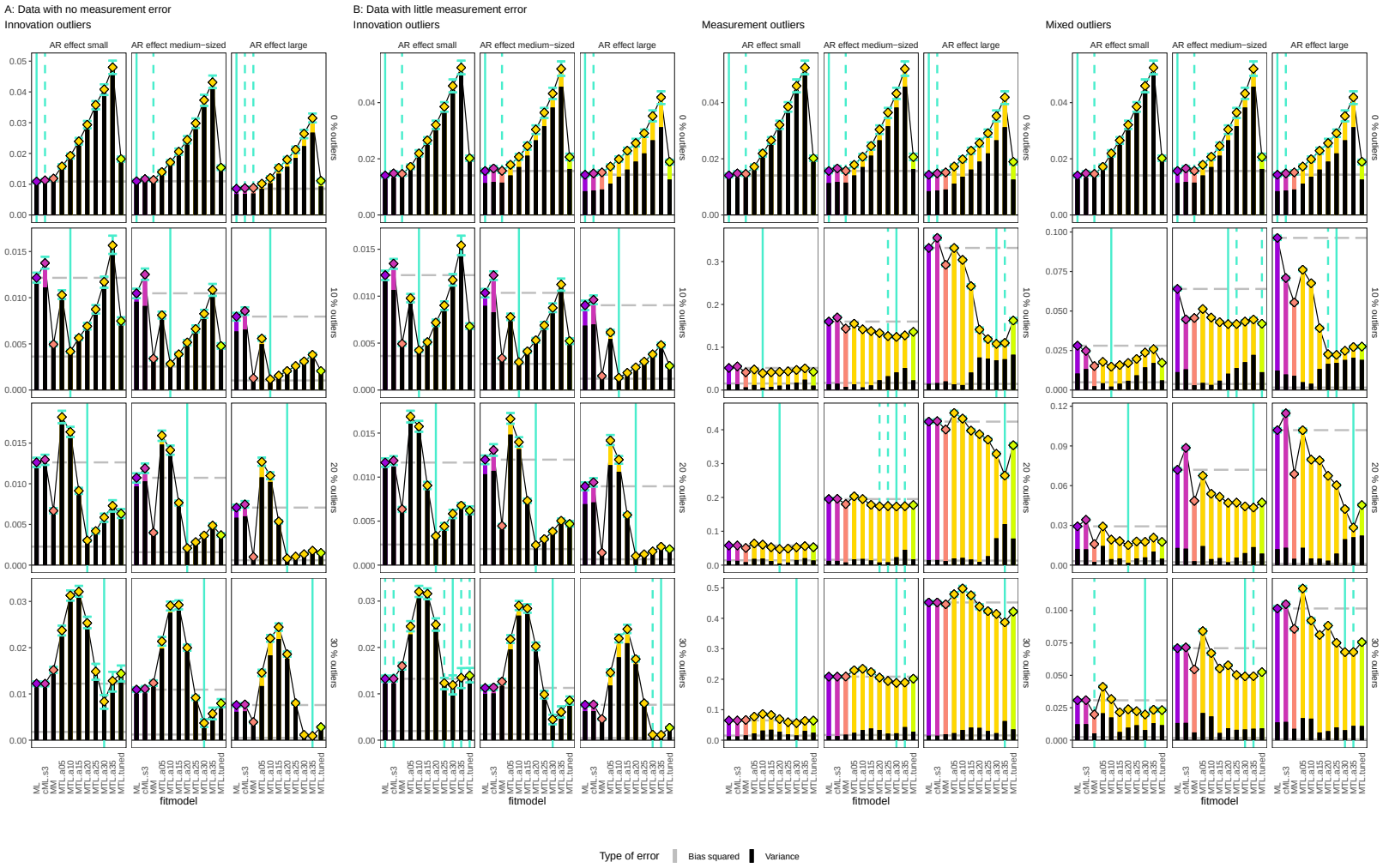


Figure 16

Estimation error AR effect for short time series ($T = 80$) with randomly occurring outliers, and models fitted without measurement error.

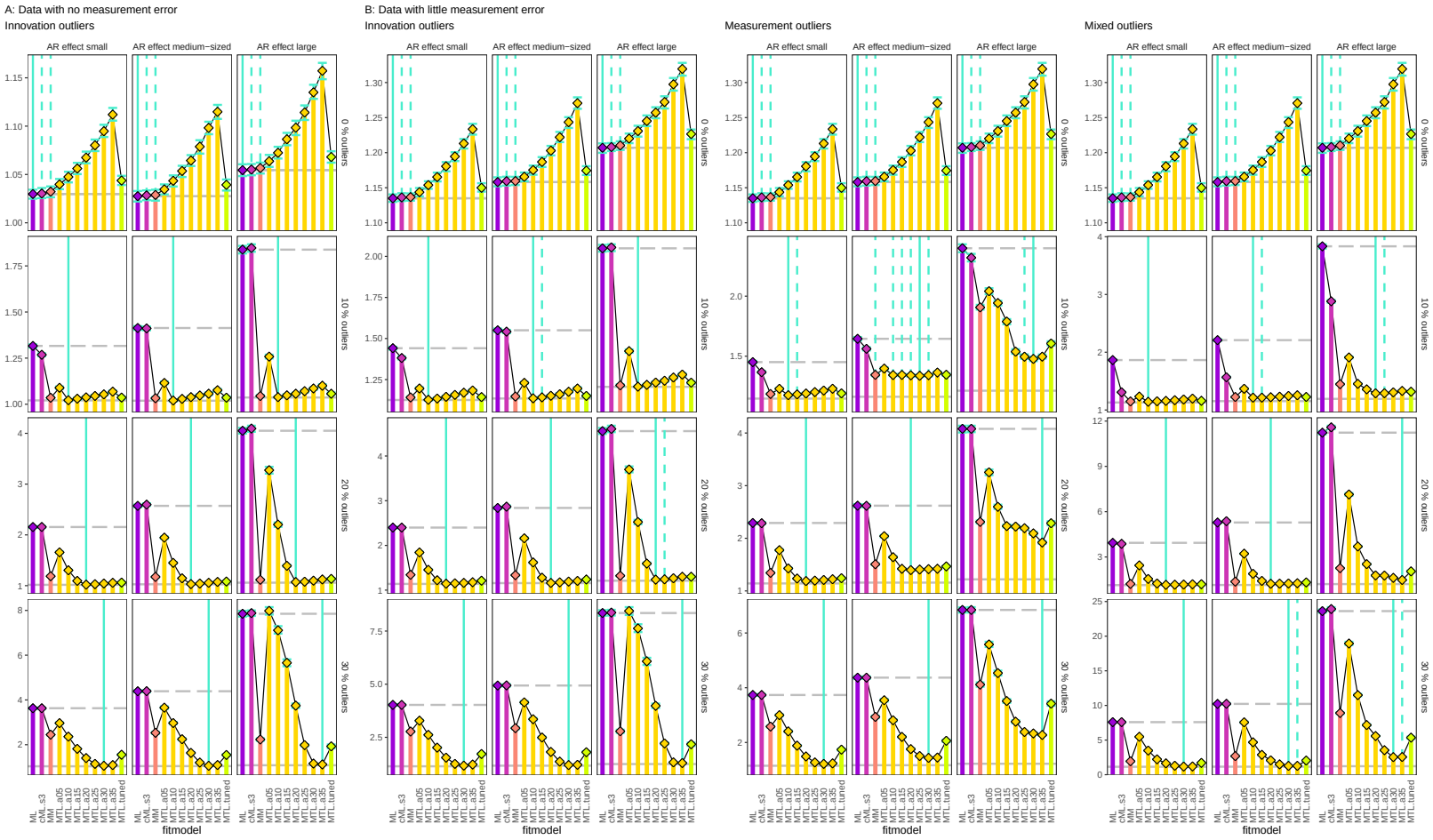


Figure 17

Prediction error for short time series ($T = 80$) with randomly occurring outliers, and models fitted without measurement error.

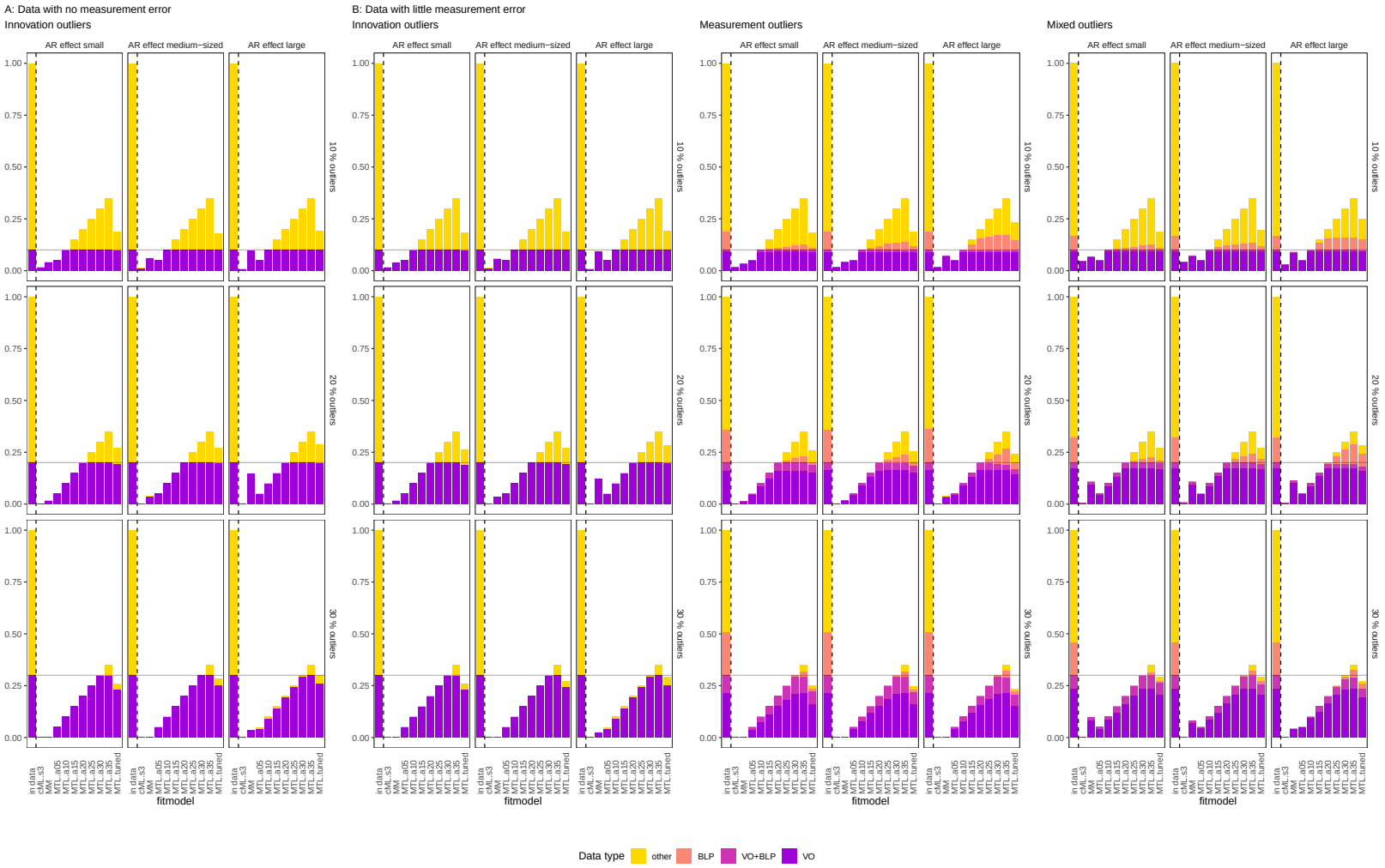


Figure 18

Relative frequencies of outliers excluded for short time series ($T = 80$) with randomly occurring outliers, and models fitted without measurement error.

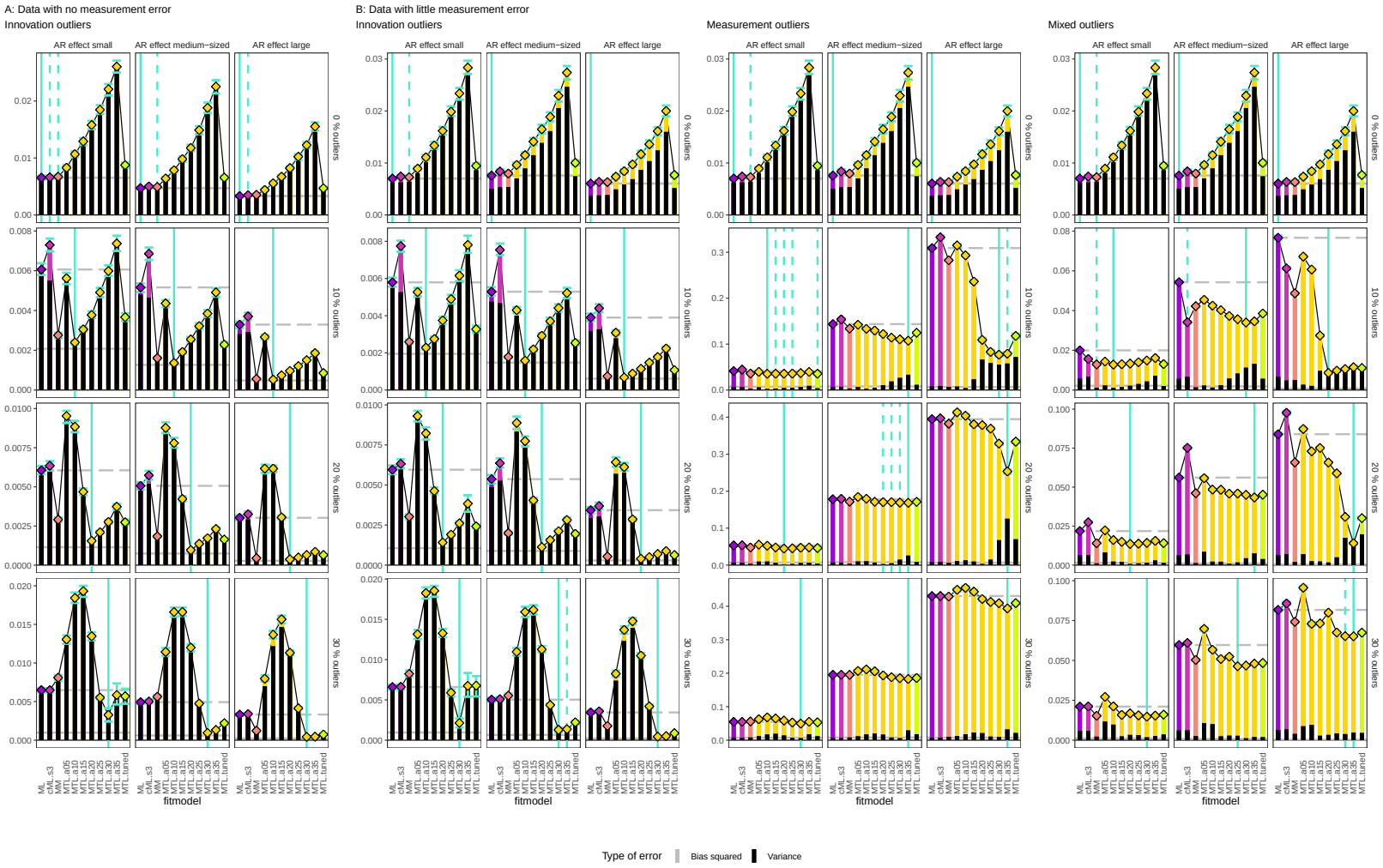


Figure 19

Estimation error AR effect for medium-length time series ($T = 160$) with randomly occurring outliers, and models fitted without measurement error.

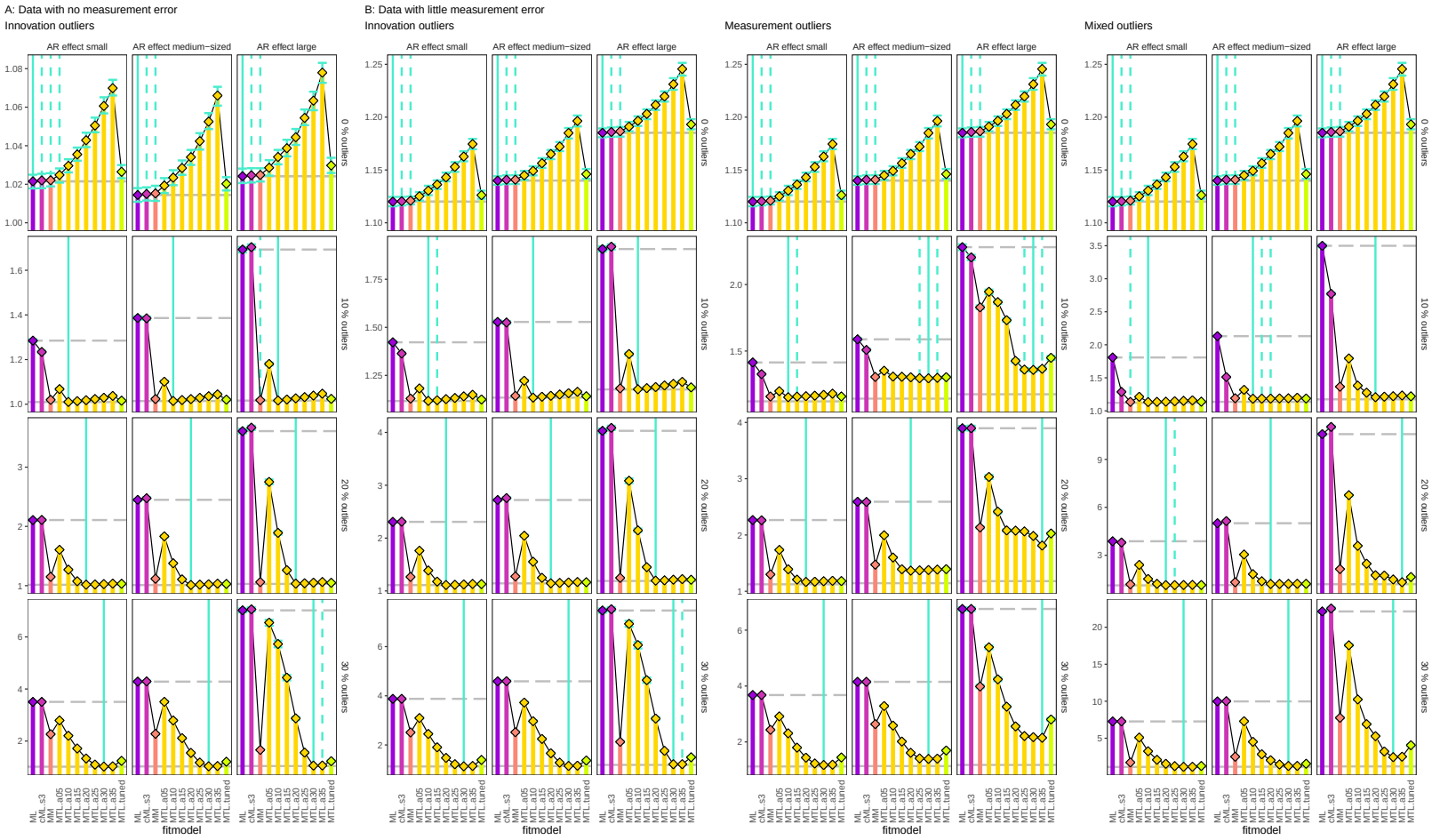


Figure 20

Prediction error for medium-length time series ($T = 160$) with randomly occurring outliers, and models fitted without measurement error.

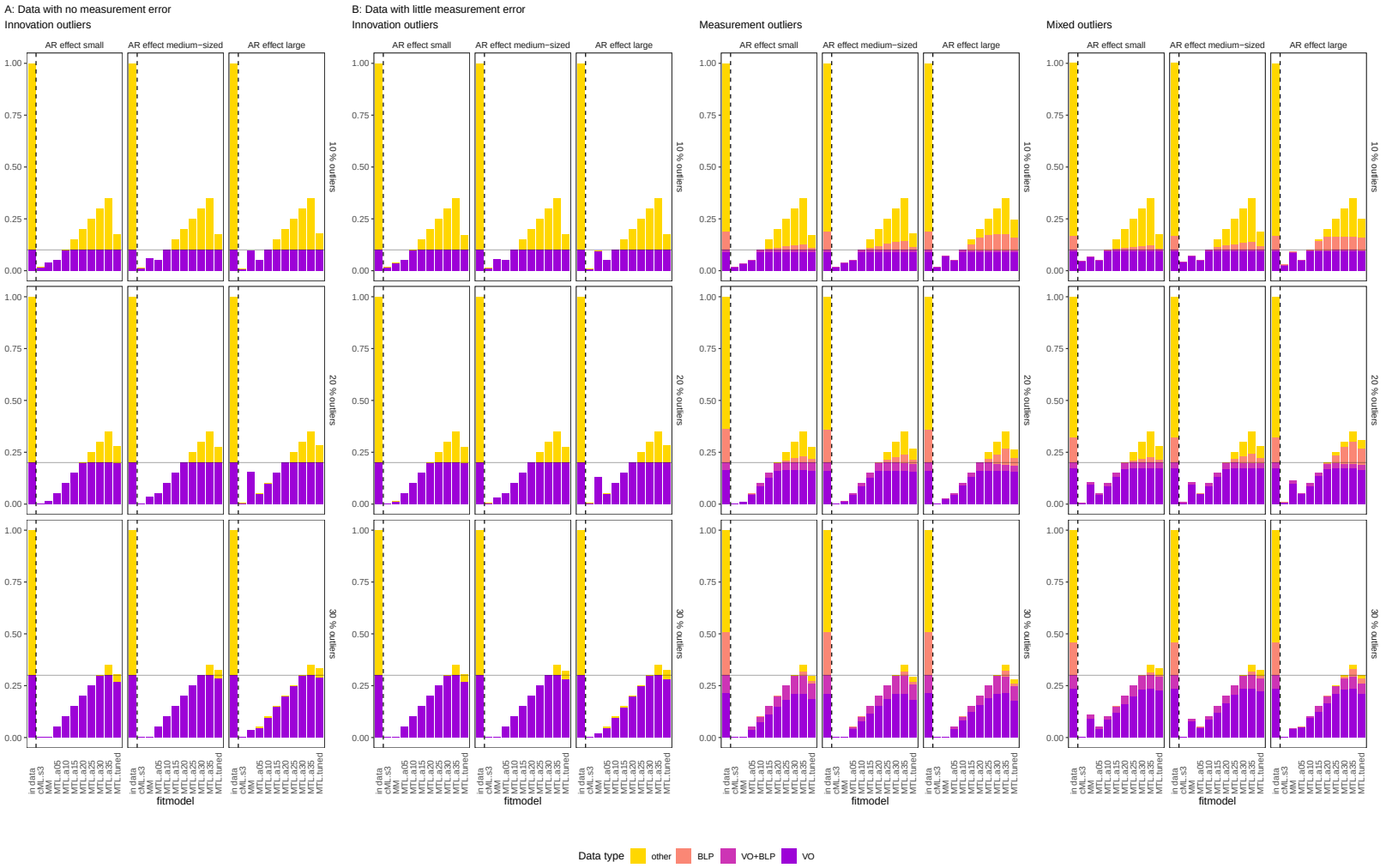


Figure 21

Relative frequencies of outliers excluded for medium-length time series ($T = 160$) with randomly occurring outliers, and models fitted without measurement error.

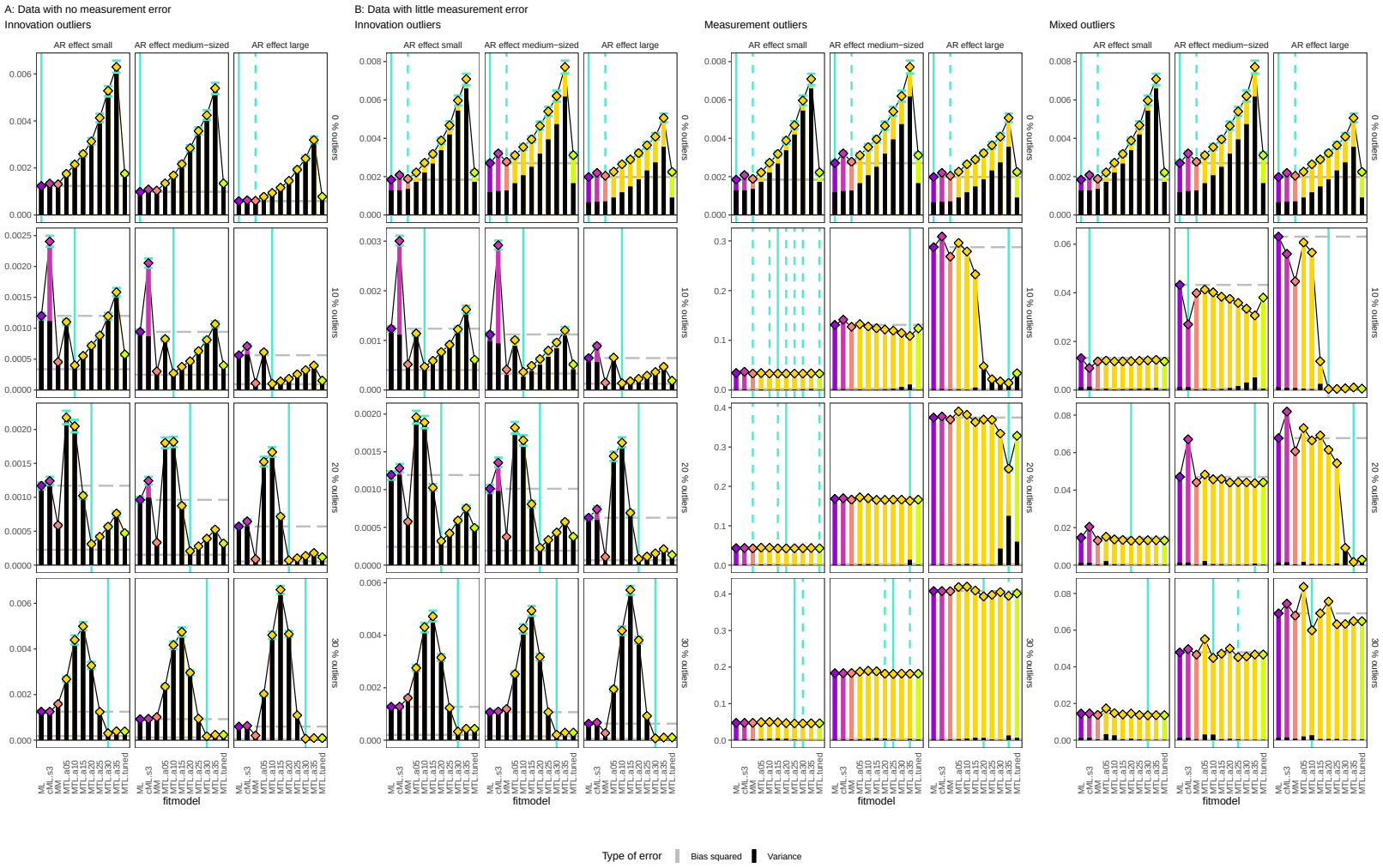


Figure 22

Estimation error AR effect for long time series ($T = 800$) with randomly occurring outliers, and models fitted without measurement error.

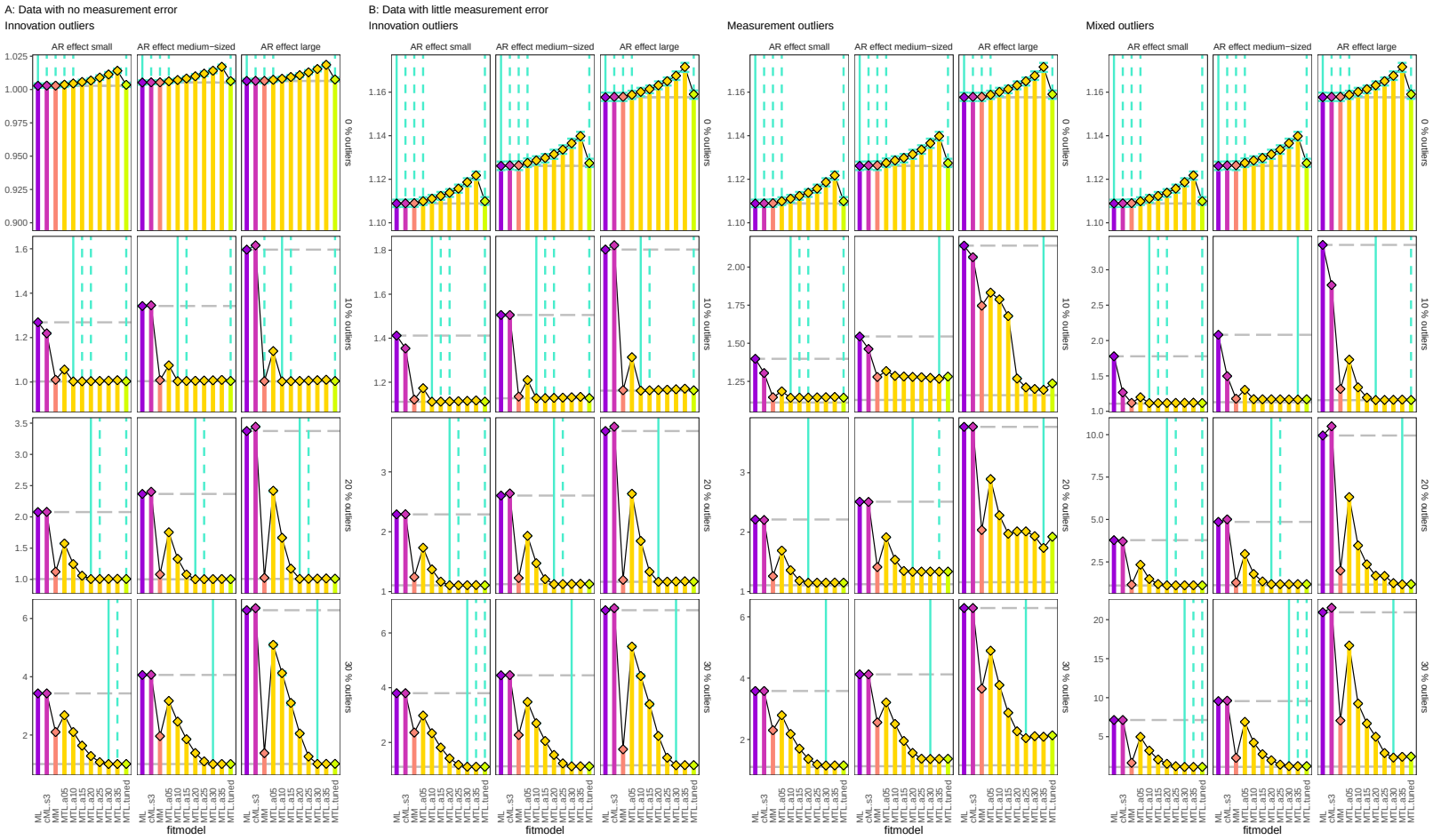


Figure 23

Prediction error for long time series ($T = 800$) with randomly occurring outliers, and models fitted without measurement error.

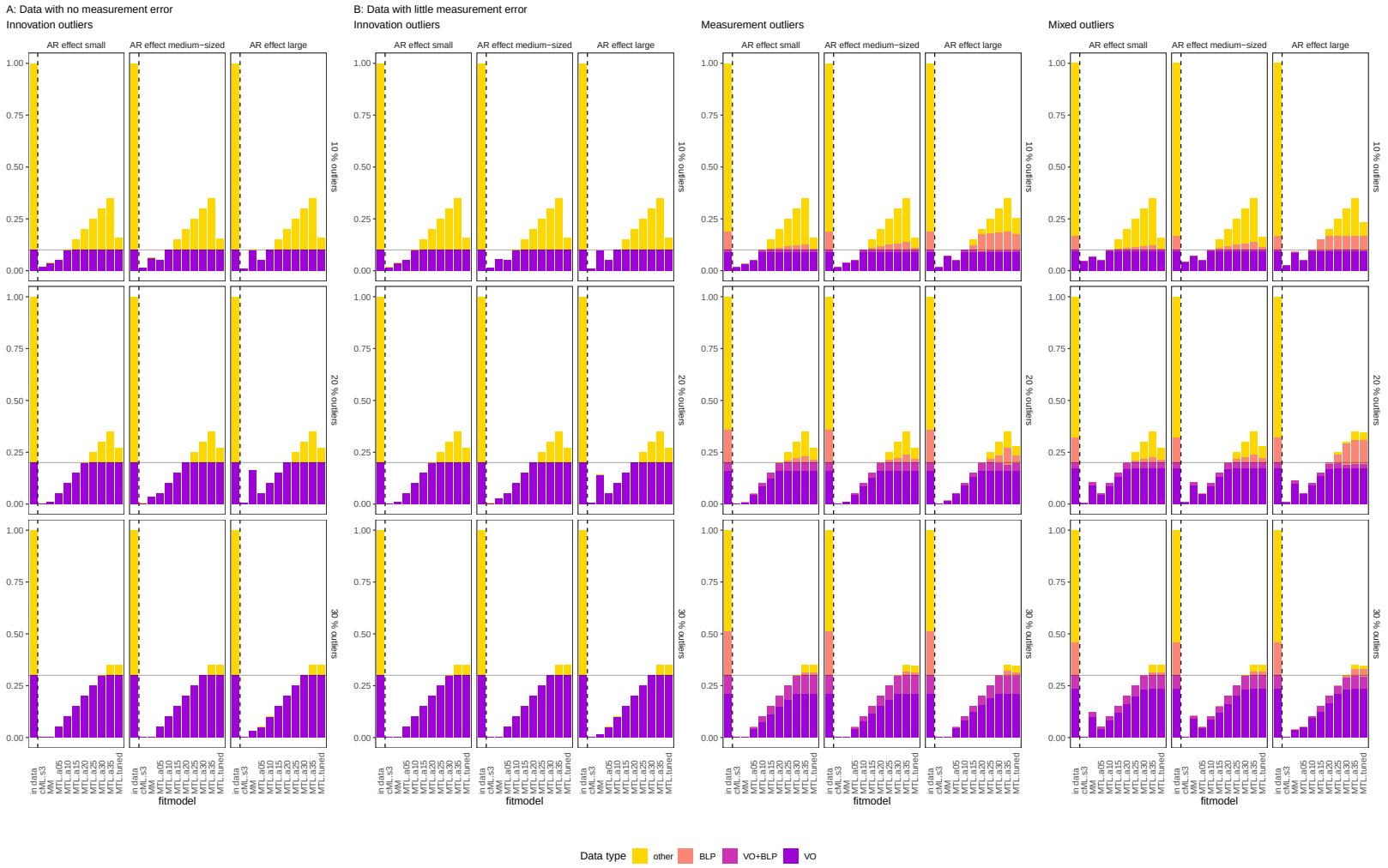


Figure 24

Relative frequencies of outliers excluded for long time series ($T = 800$) with randomly occurring outliers, and models fitted without measurement error.

2.2.4 Improved estimation and prediction due to robust methods

The figures in this section display performance of the different robust methods under different types of outliers. Unlike in section 2.2.4 this section not only includes the $3-\sigma$ cML estimator, but also the alternative cML variants.

Figures 25, 26, 31, 32, 37, and 38 show the estimation error of the AR effect for short, medium-length and long time series, respectively. Figures 27, 28, 33, 34, 39, and 40 show the prediction error for short, medium-length and long time series, respectively. Figures 29, 30, 35, 36, 41, and 42 show the outlier handling for short, medium-length and long time series, respectively.

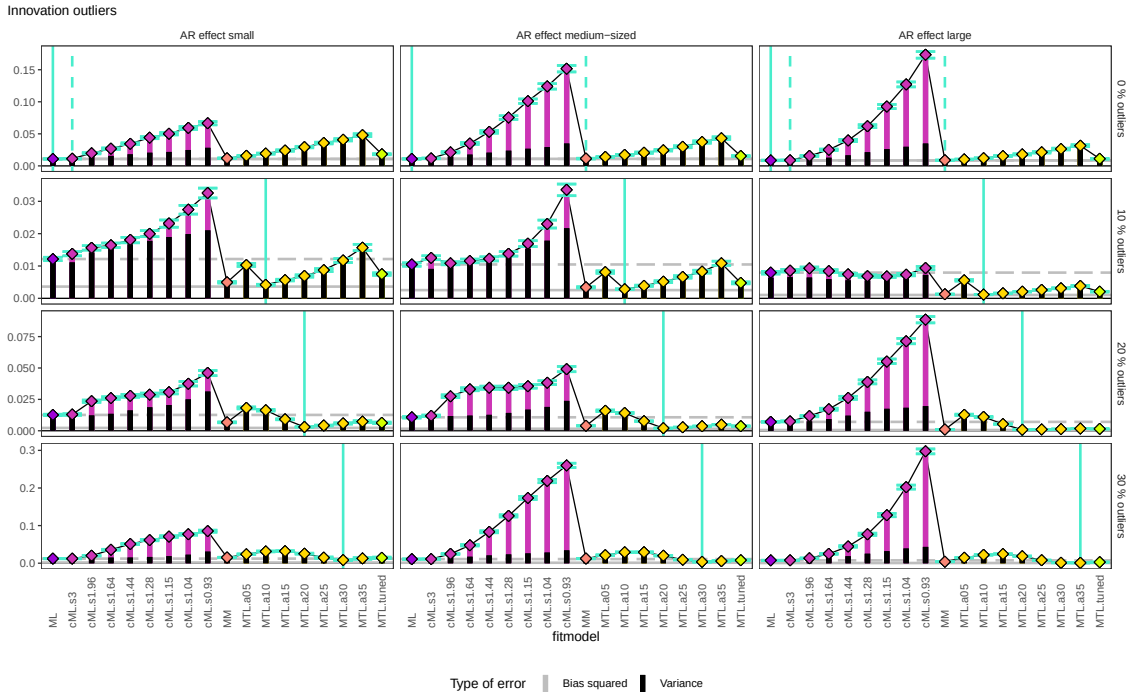
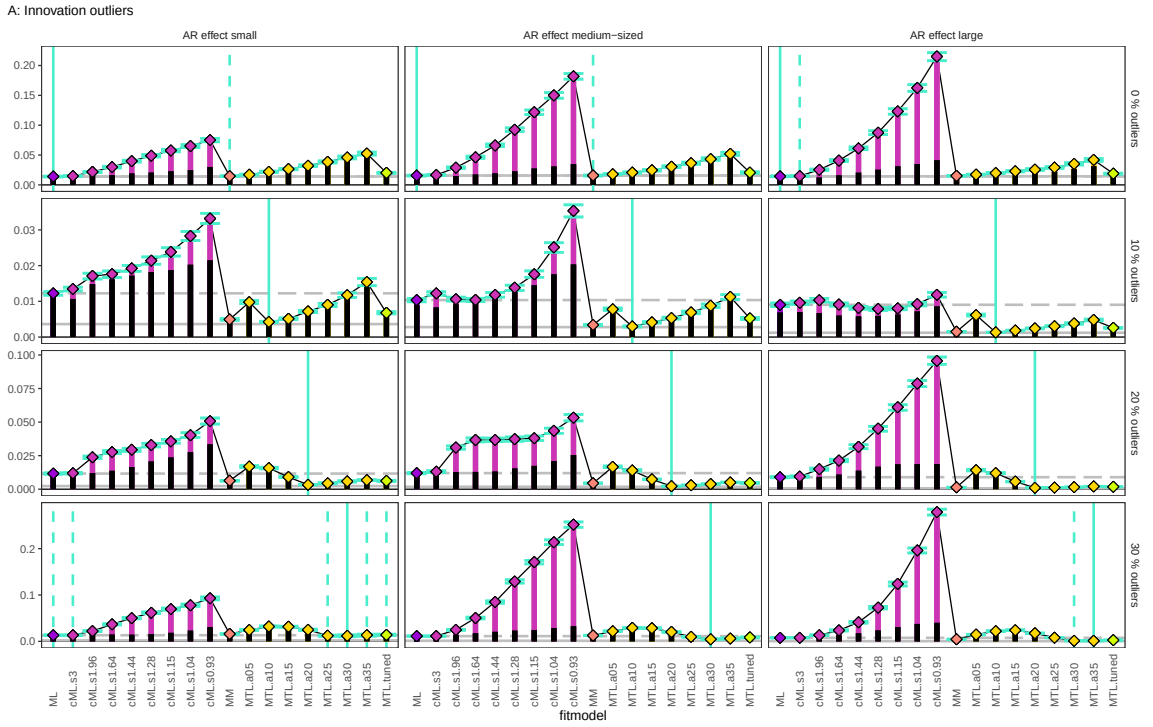


Figure 25

Estimation error AR effect for data with no measurement error for short time series ($T = 80$) with randomly occurring outliers, and models fitted without measurement error.



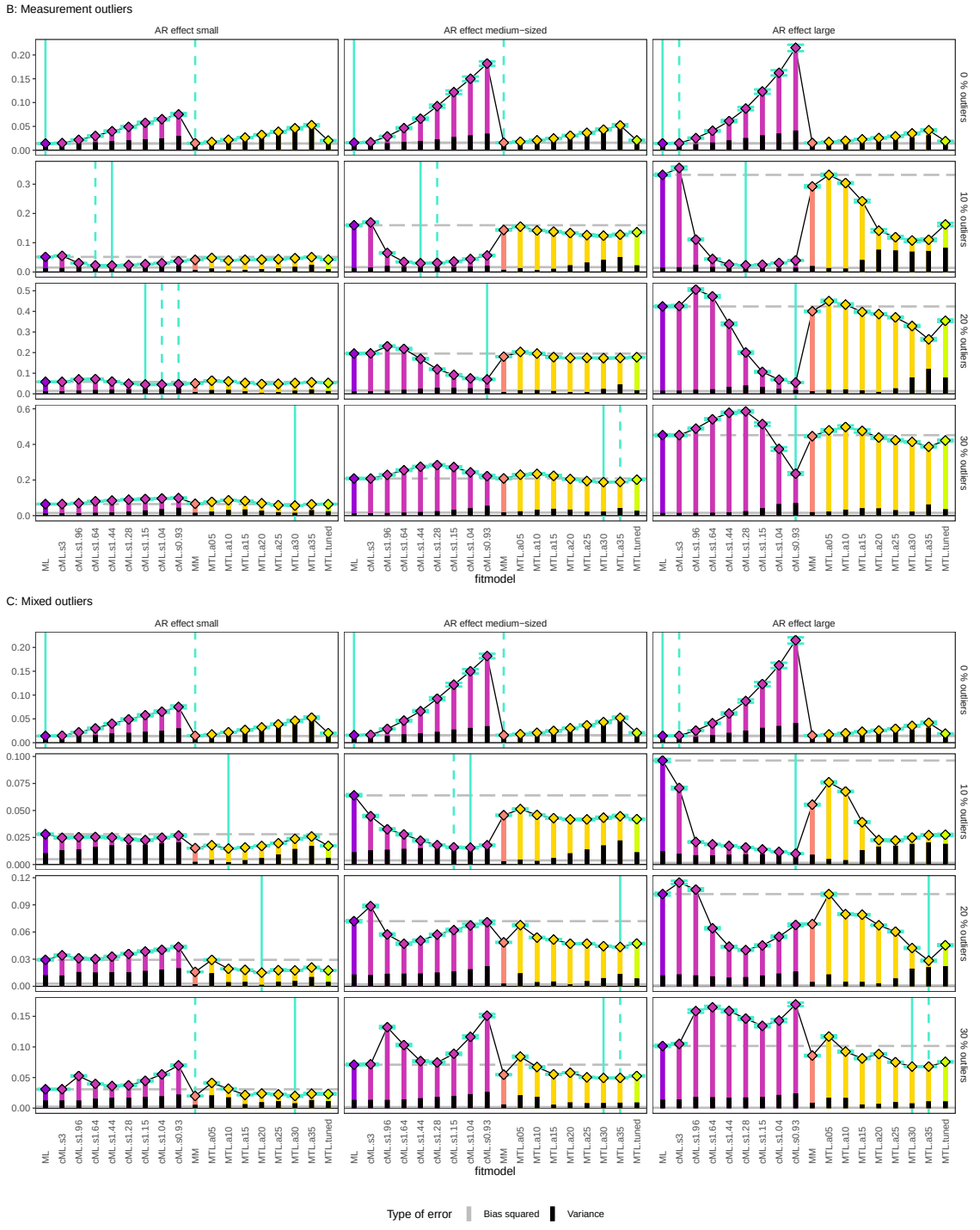


Figure 26

Estimation error AR effect for data with little measurement error for short time series ($T = 80$) with randomly occurring outliers, and models fitted without measurement error.

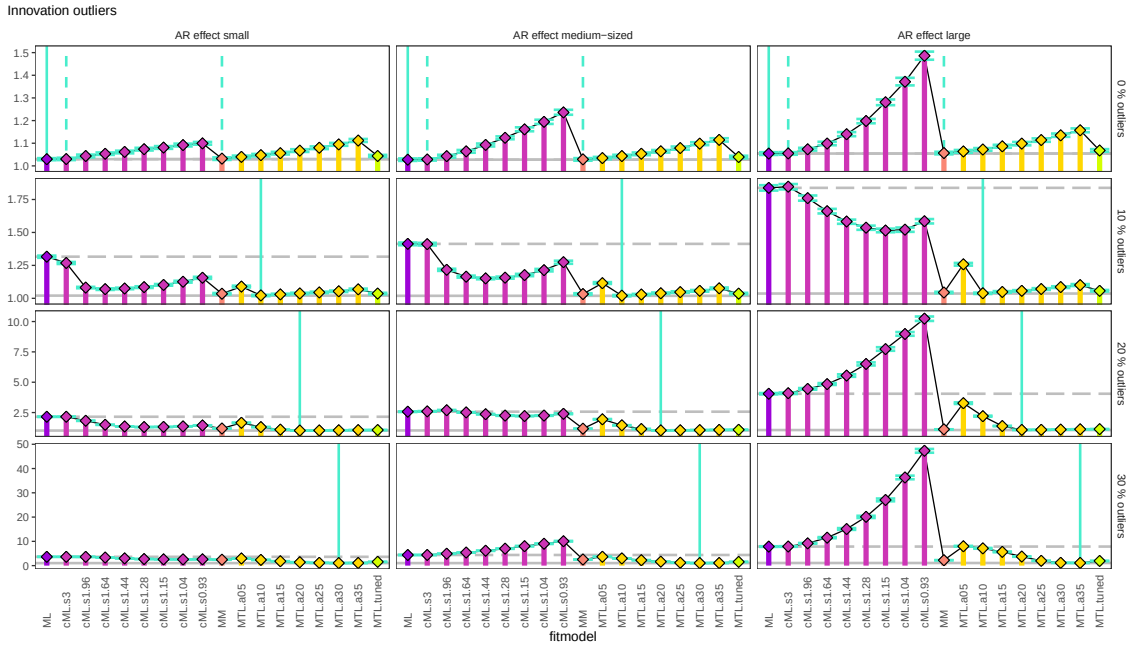
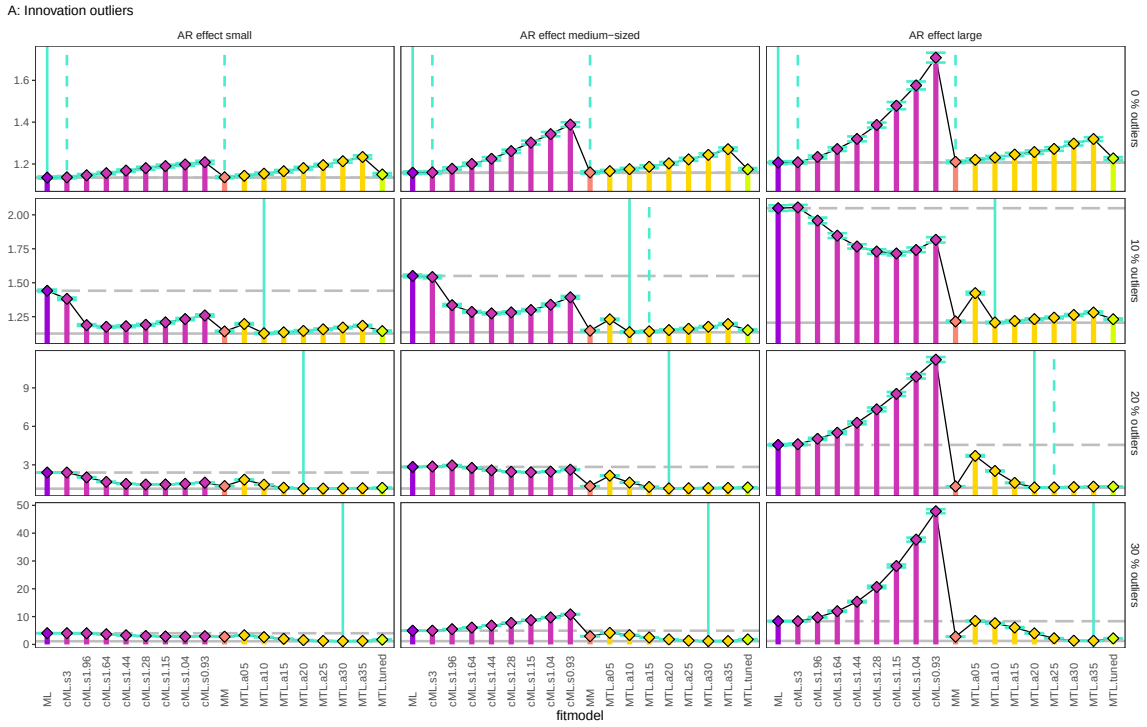


Figure 27

Prediction error for data with no measurement error for short time series ($T = 80$) with randomly occurring outliers, and models fitted without measurement error.



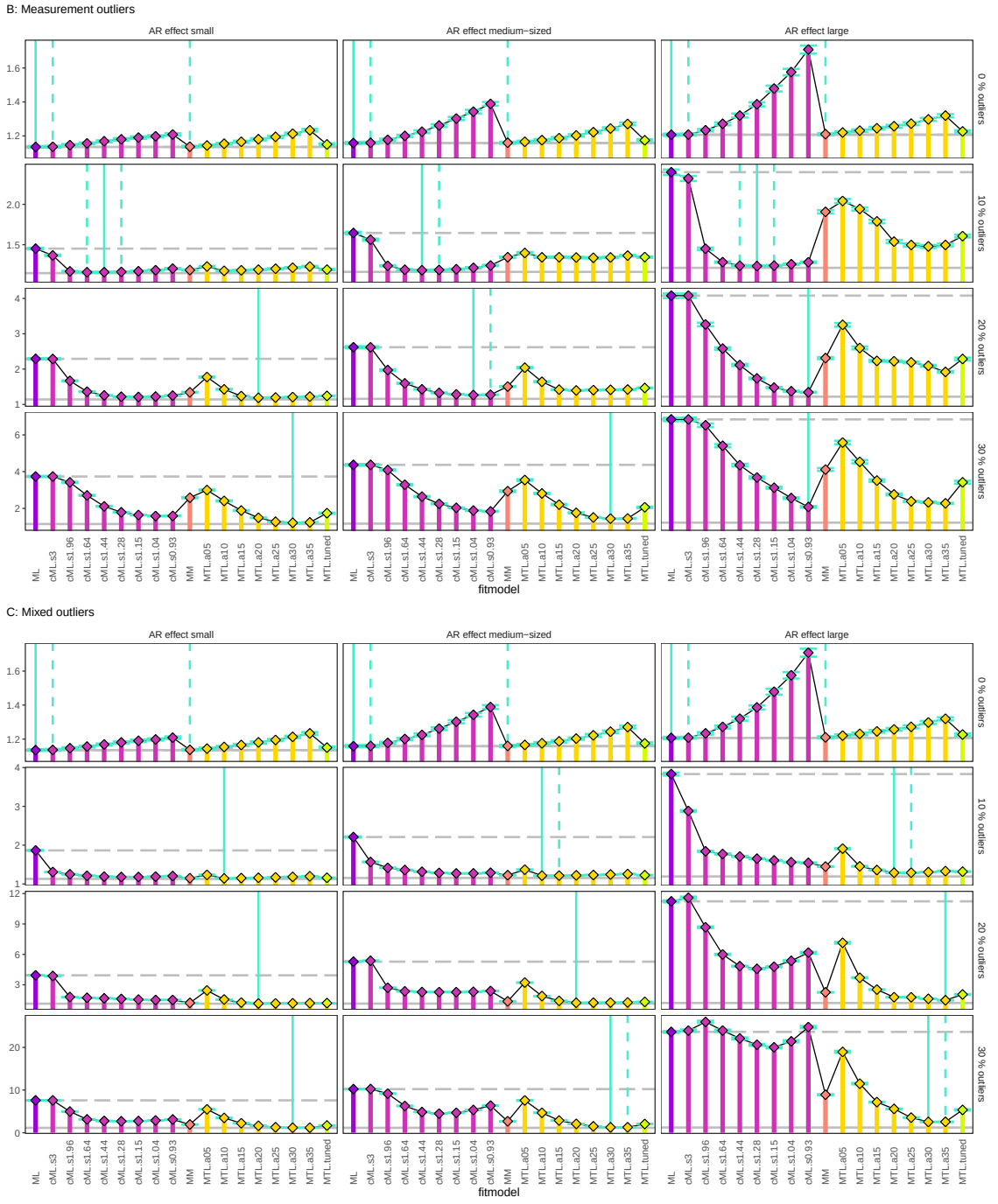


Figure 28

Prediction error for data with little measurement error for short time series ($T = 80$) with randomly occurring outliers, and models fitted without measurement error.

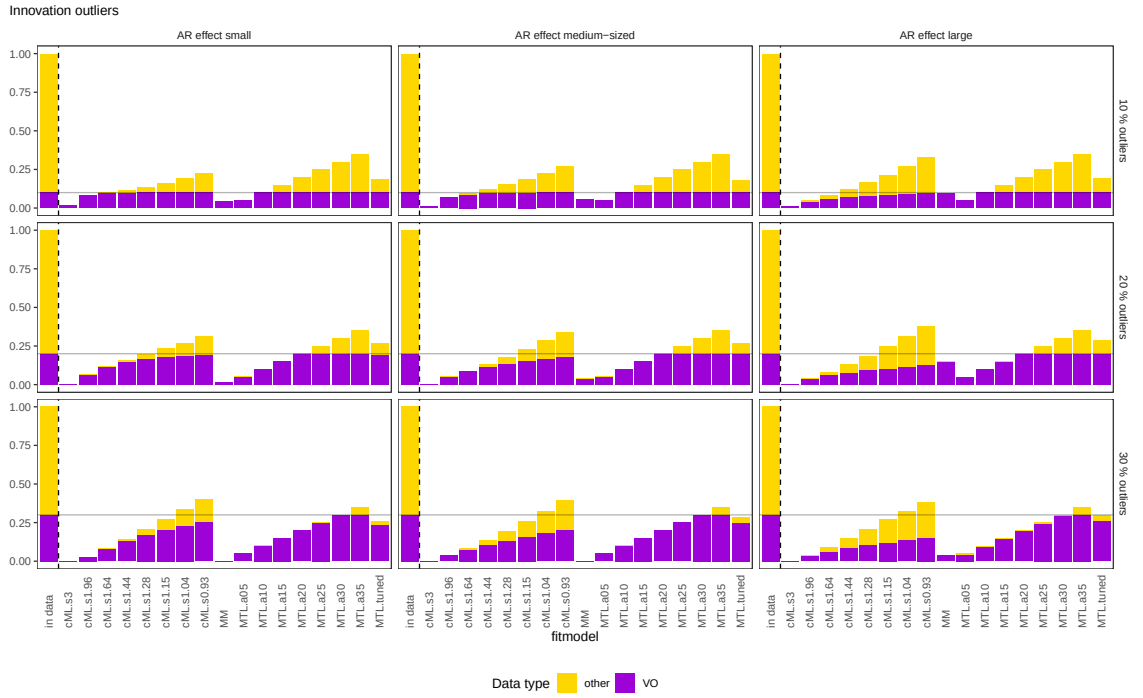
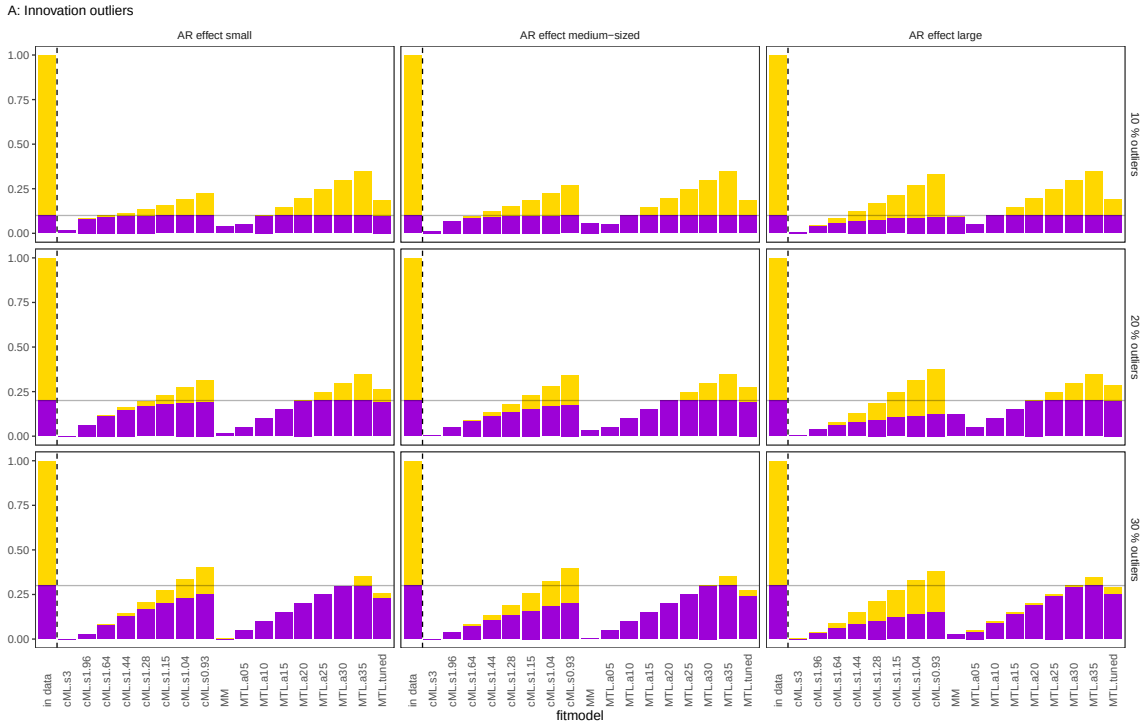


Figure 29

Relative frequencies of outliers excluded for data with no measurement error for short time series ($T = 80$) with randomly occurring outliers, and models fitted without measurement error.



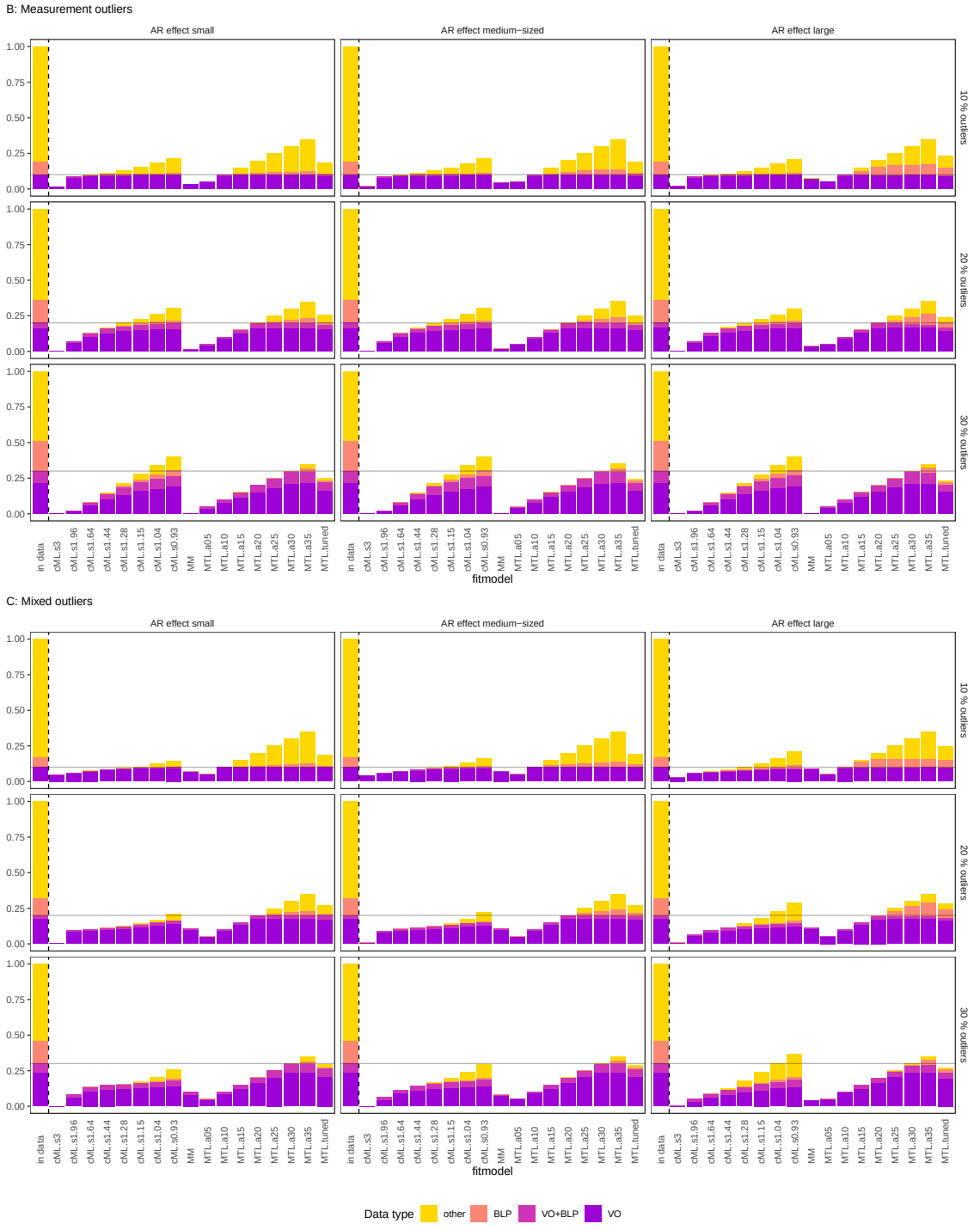


Figure 30

Relative frequencies of outliers excluded for data with little measurement error for short time series ($T = 80$) with randomly occurring outliers, and models fitted without measurement error.

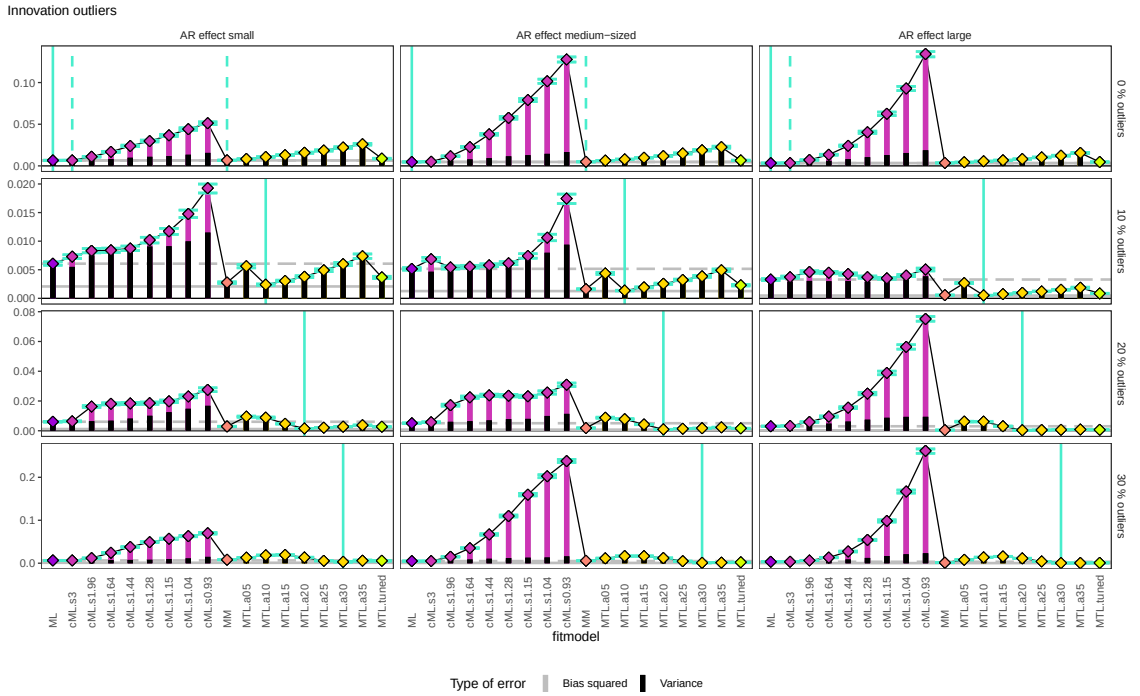
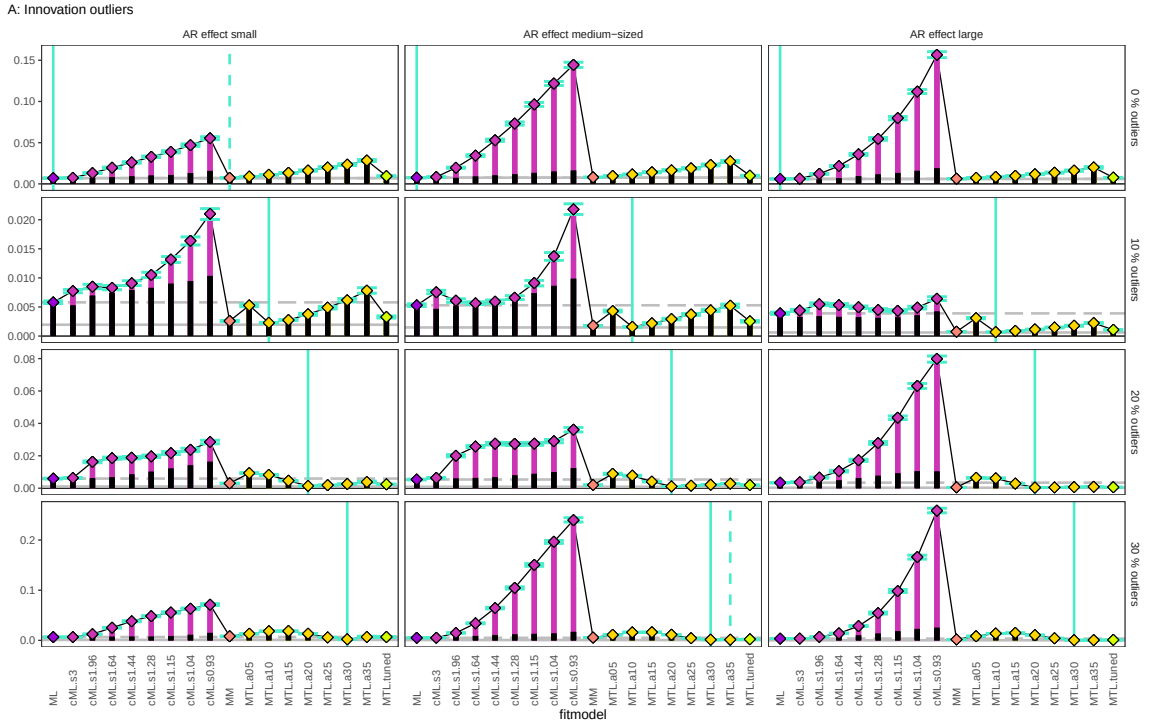


Figure 31

Estimation error AR effect for data with no measurement error for medium-length time series ($T = 160$) with randomly occurring outliers, and models fitted without measurement error.



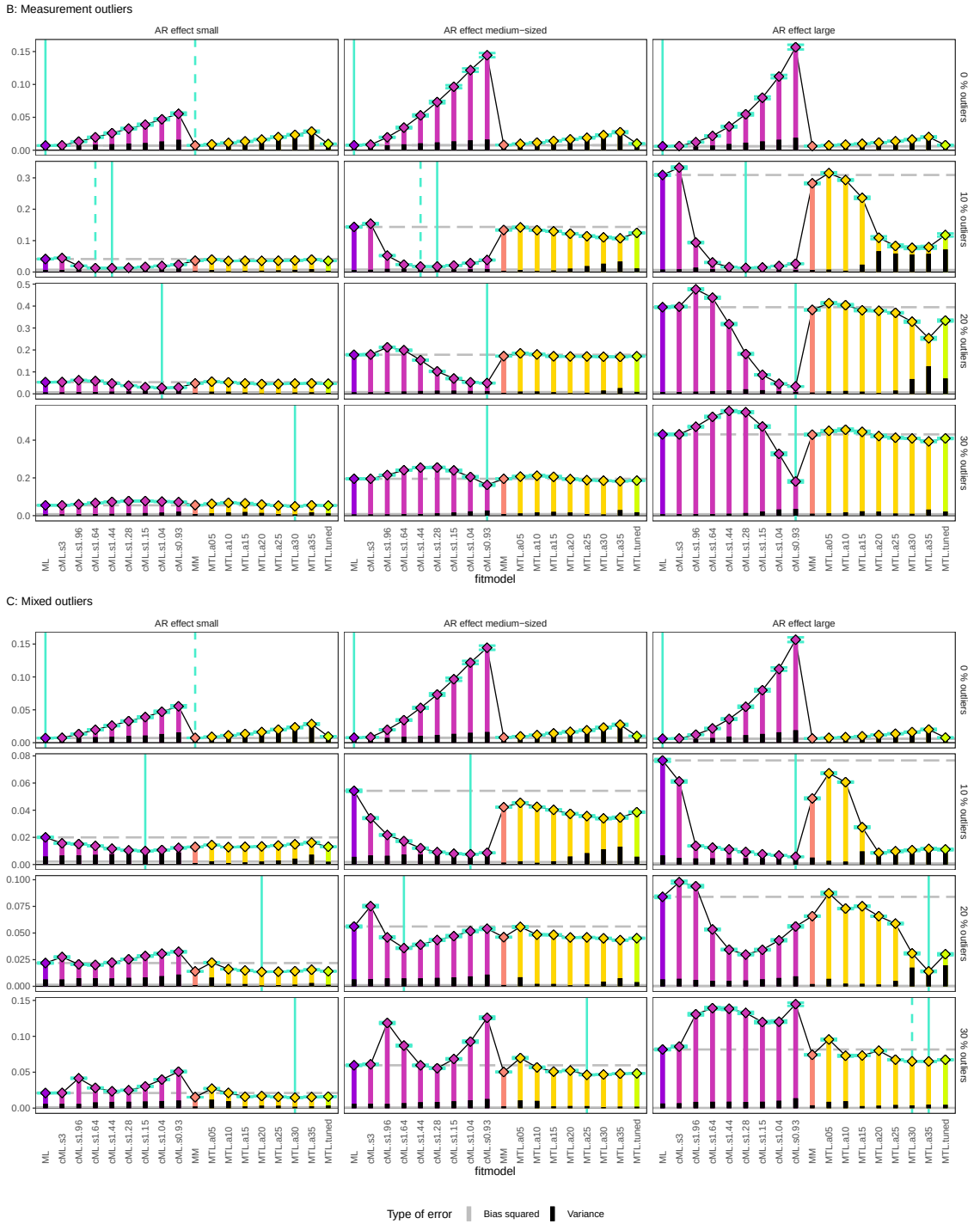


Figure 32

Estimation error AR effect for data with little measurement error for medium-length time series ($T = 160$) with randomly occurring outliers, and models fitted without measurement error.

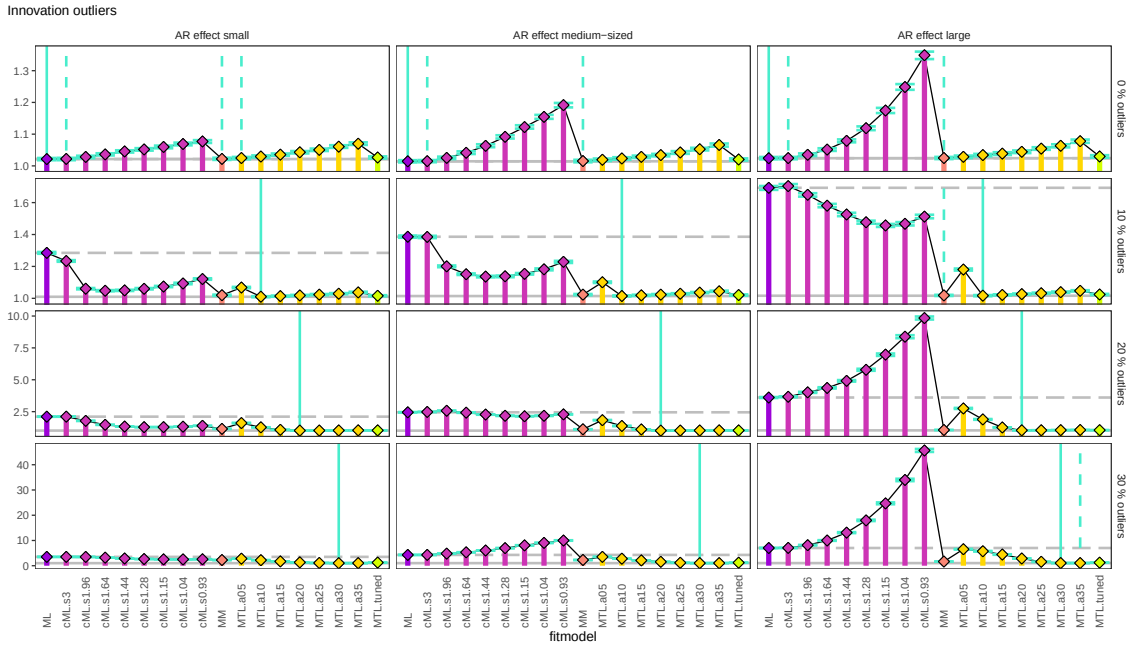


Figure 33

Prediction error for data with no measurement error for medium-length time series ($T = 160$) with randomly occurring outliers, and models fitted without measurement error.

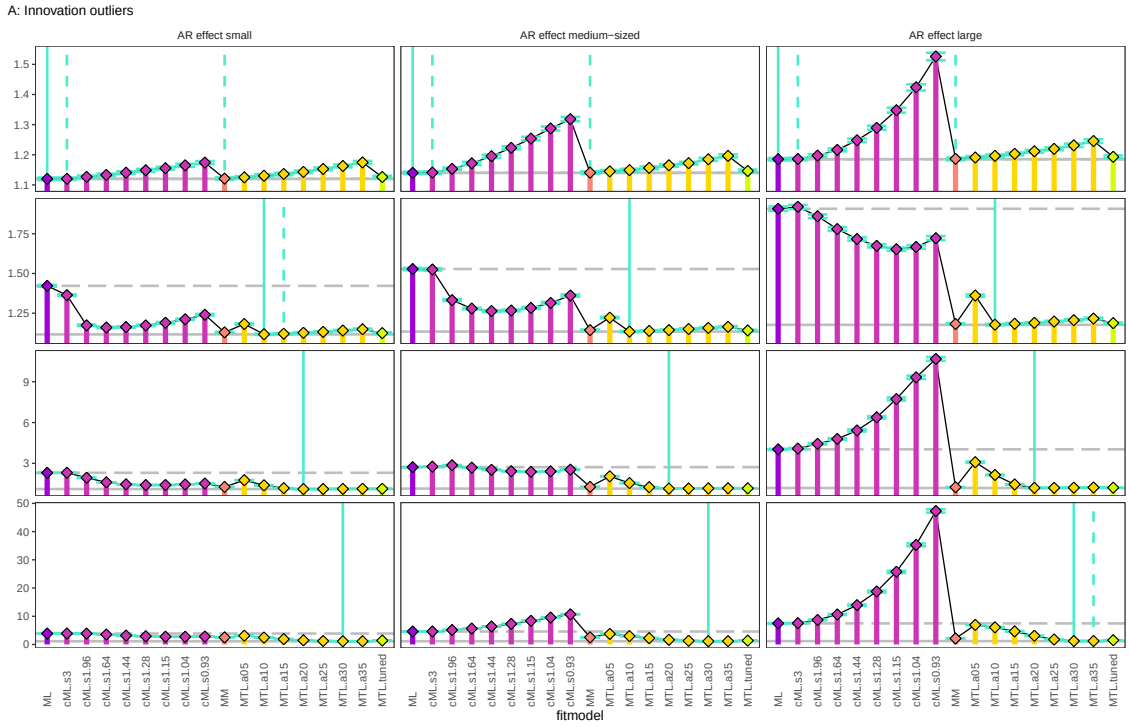


Figure 34

Prediction error for data with little measurement error for medium-length time series ($T = 160$) with randomly occurring outliers, and models fitted without measurement error.

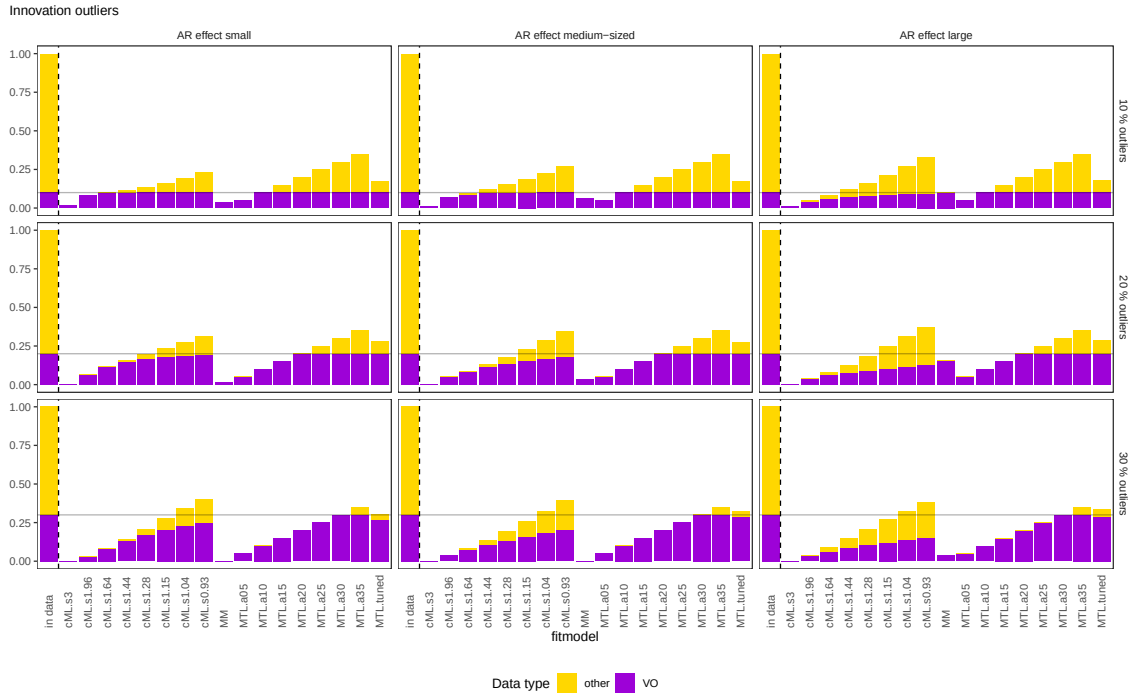
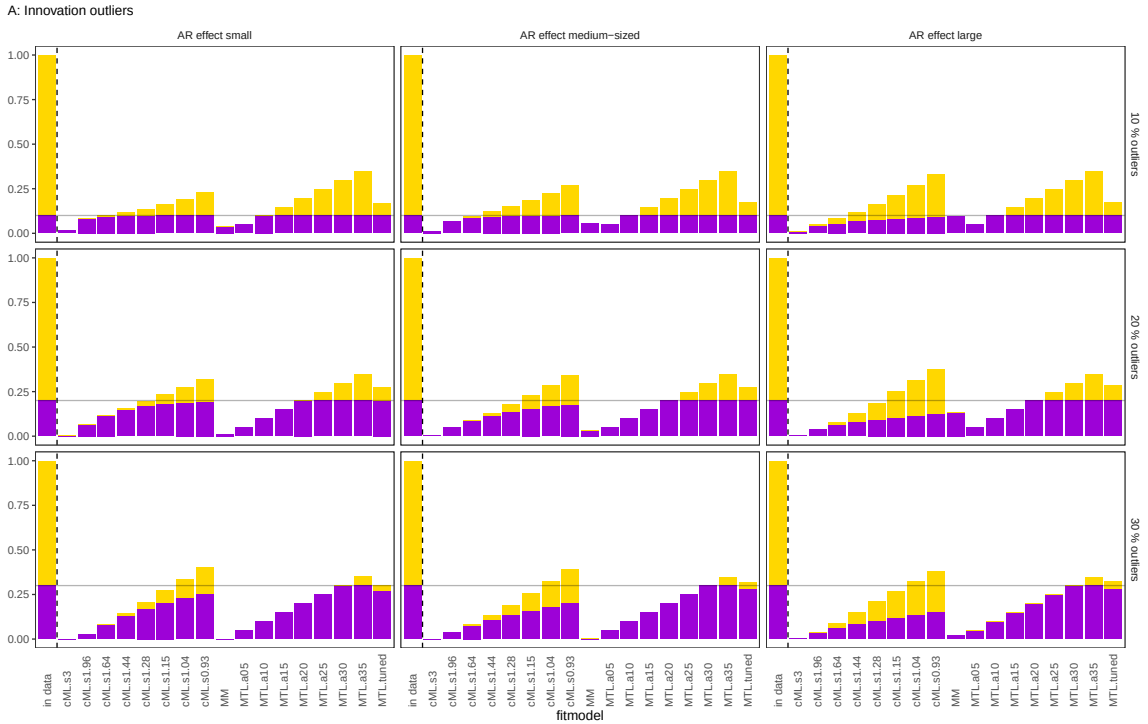


Figure 35

Relative frequencies of outliers excluded for data with no measurement error for medium-length time series ($T = 160$) with randomly occurring outliers, and models fitted without measurement error.



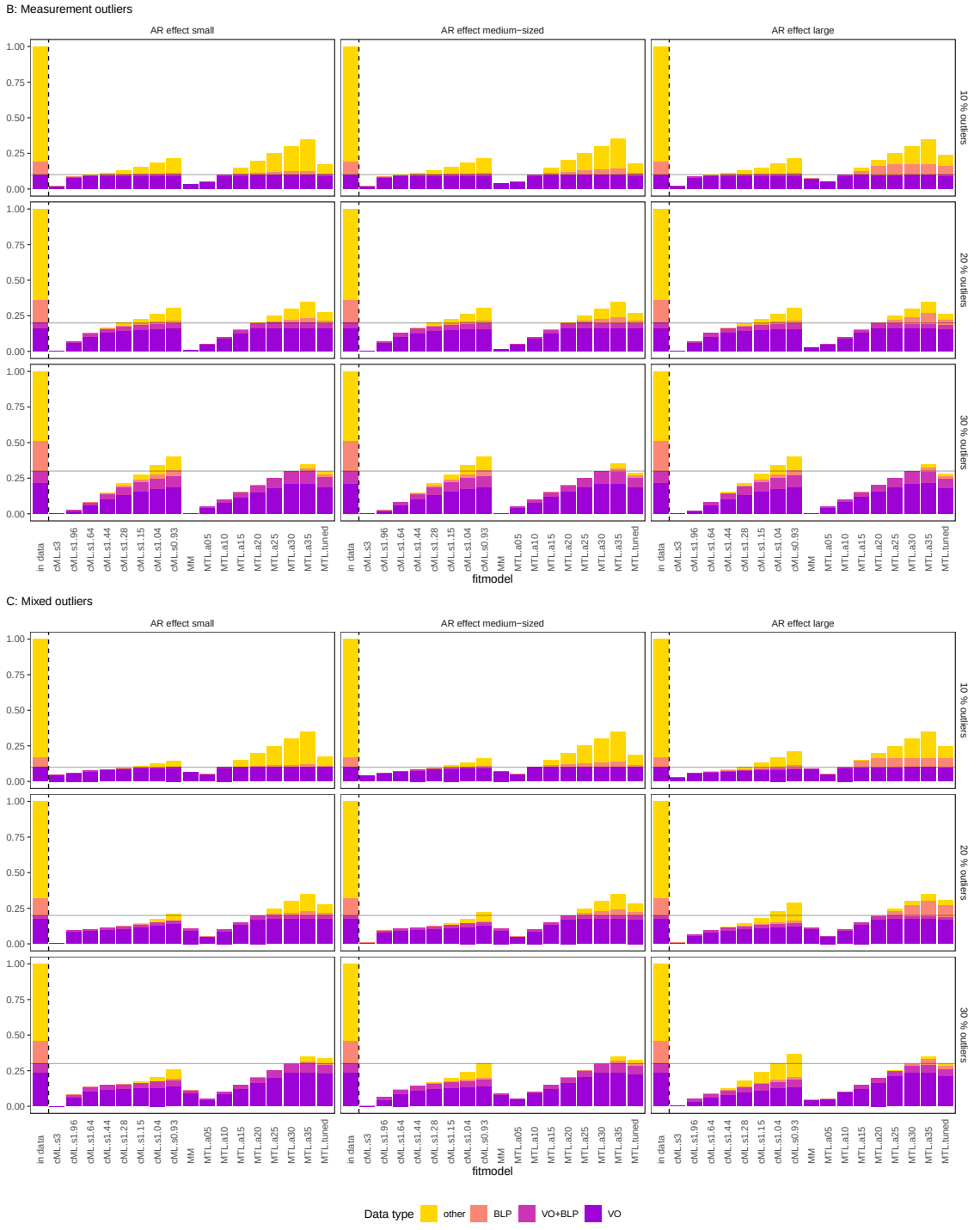


Figure 36

Relative frequencies of outliers excluded for data with little measurement error for medium-length time series ($T = 160$) with randomly occurring outliers, and models fitted without measurement error.

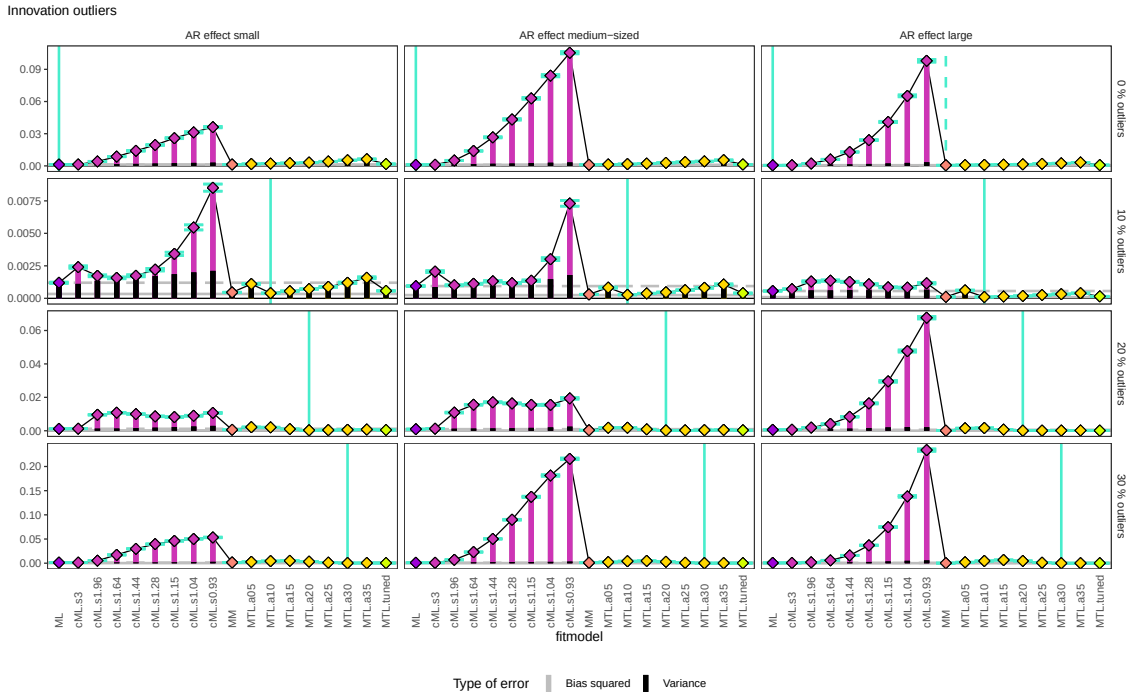
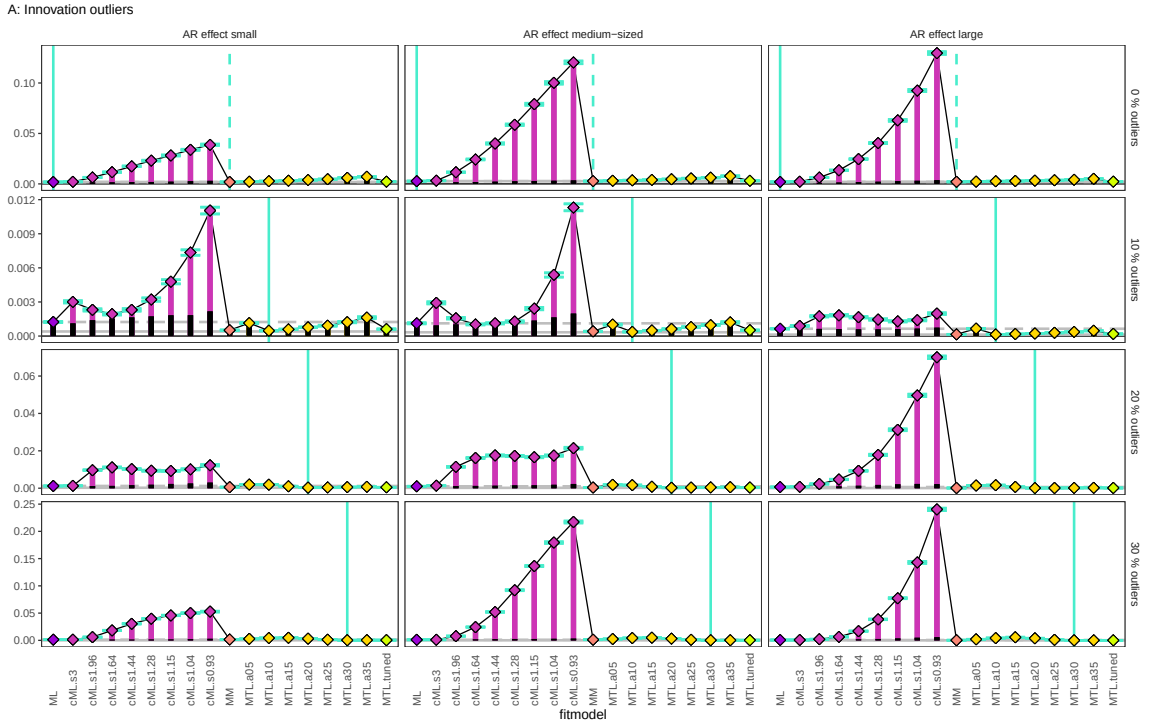


Figure 37

Estimation error AR effect for data with no measurement error for long time series ($T = 800$) with randomly occurring outliers, and models fitted without measurement error.



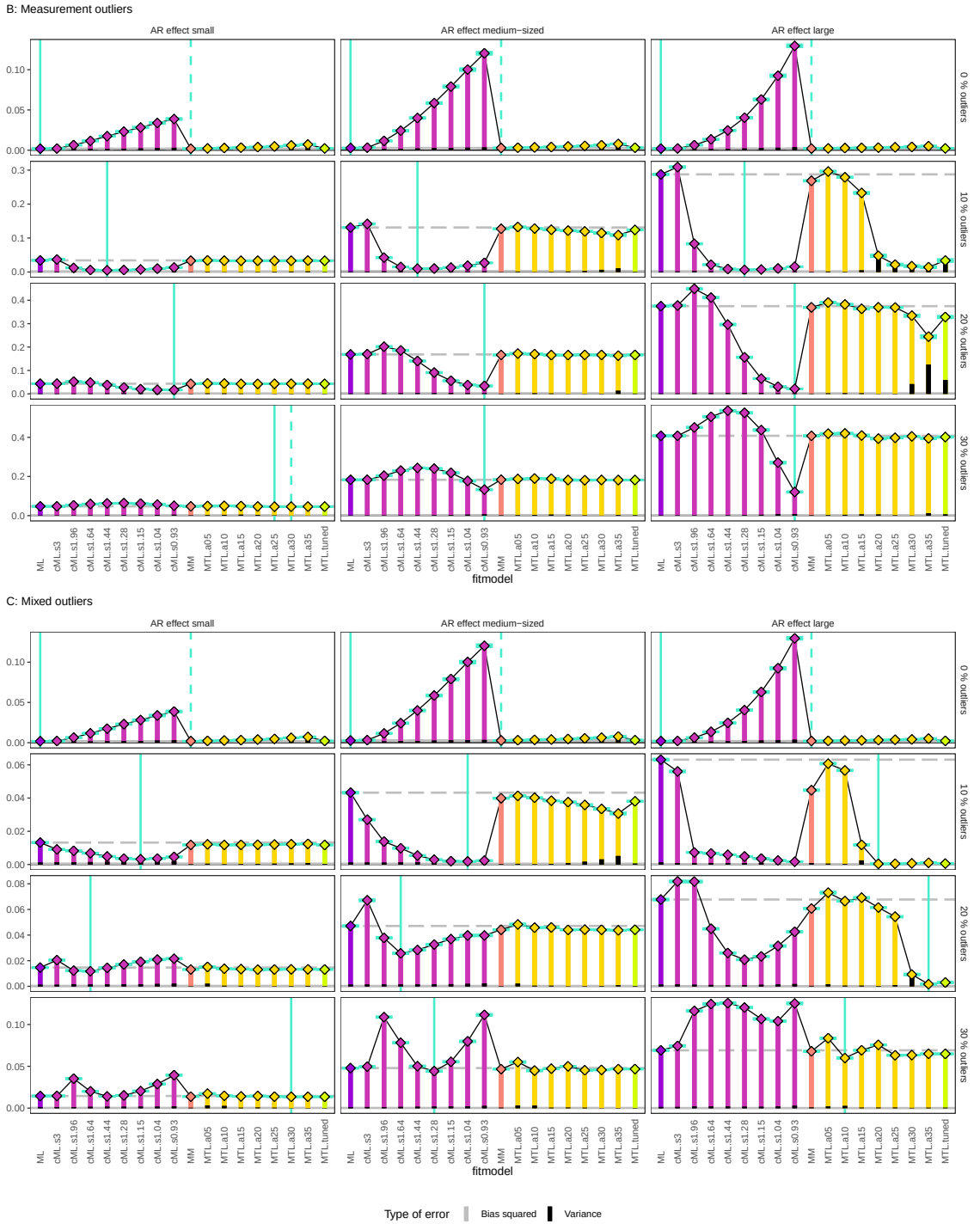


Figure 38

Estimation error AR effect for data with little measurement error for long time series ($T = 800$) with randomly occurring outliers, and models fitted without measurement error.

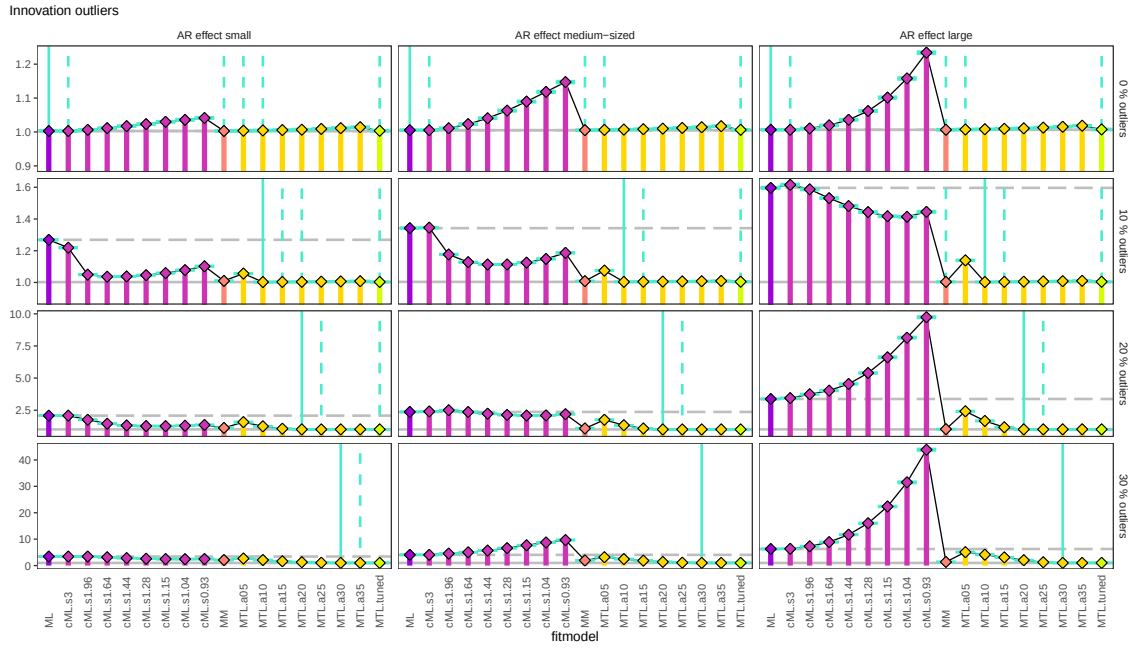
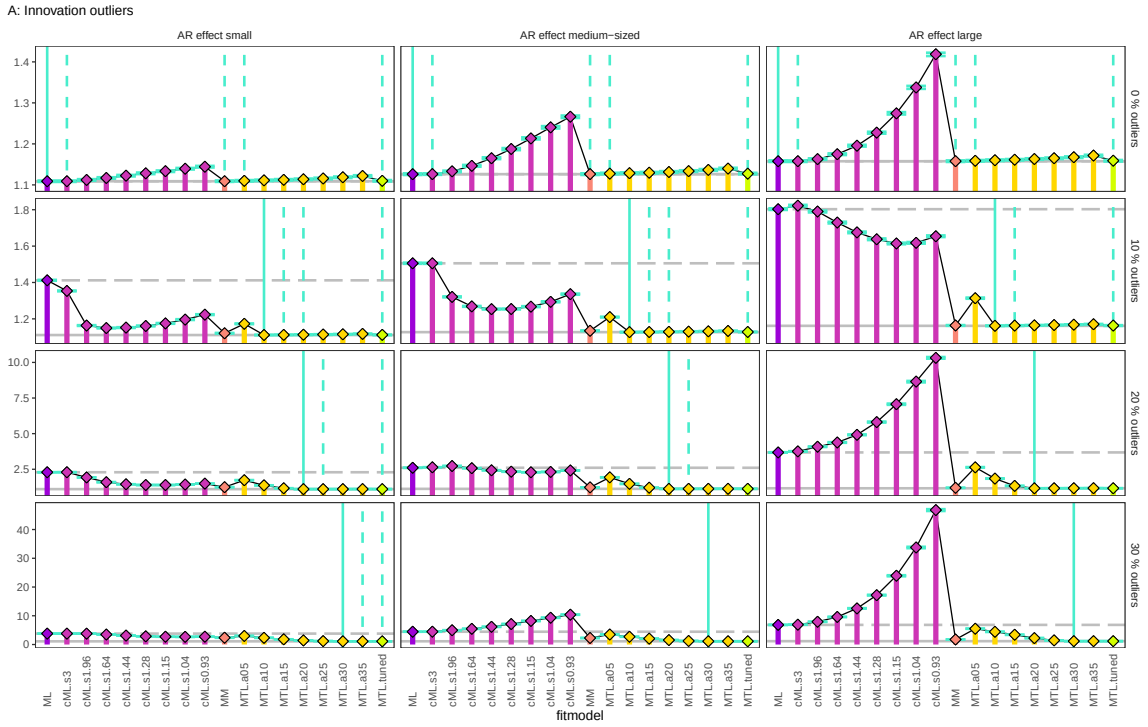


Figure 39

Prediction error for data with no measurement error for long time series ($T = 800$) with randomly occurring outliers, and models fitted without measurement error.



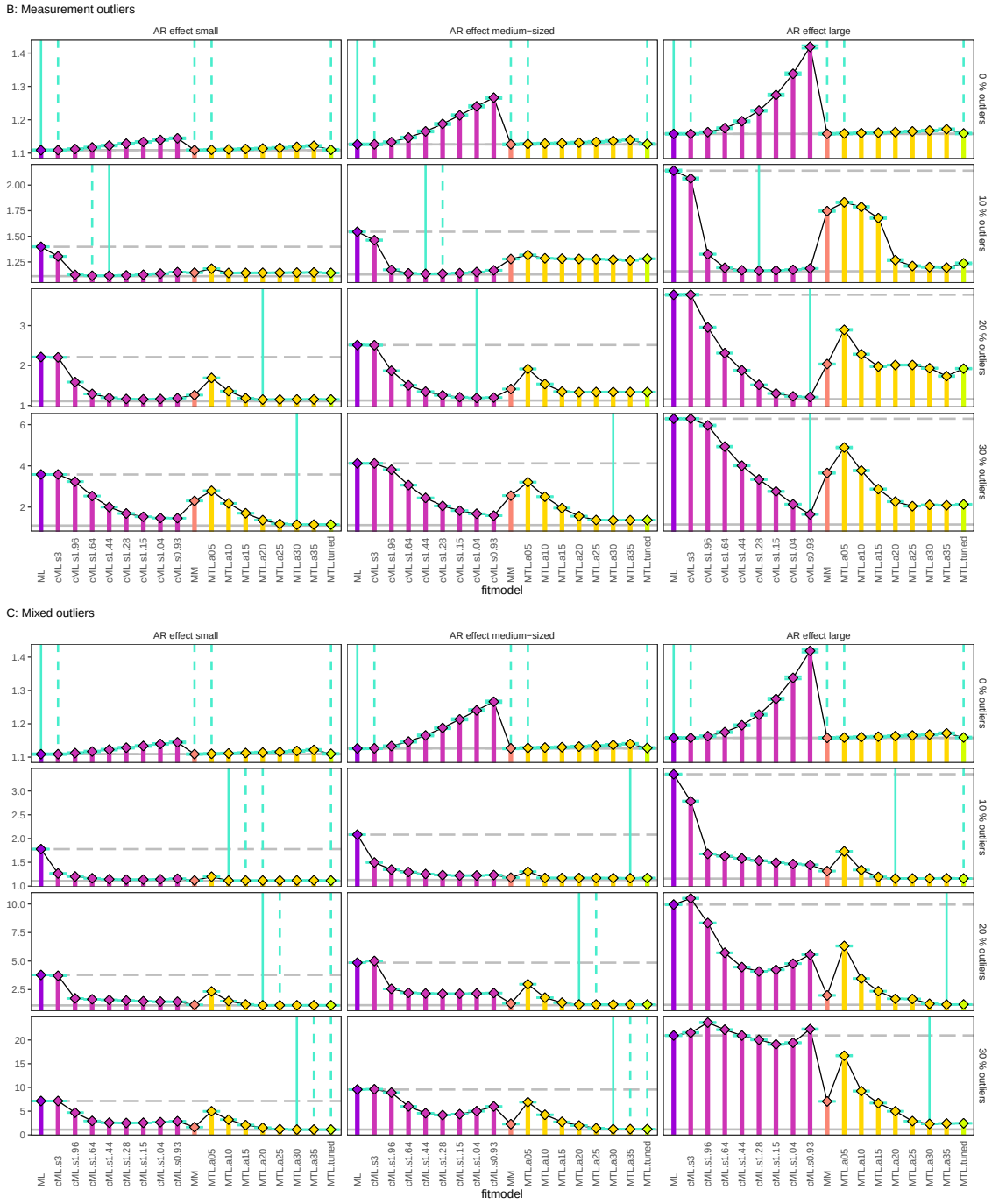


Figure 40

Prediction error for data with little measurement error for long time series ($T = 800$) with randomly occurring outliers, and models fitted without measurement error.

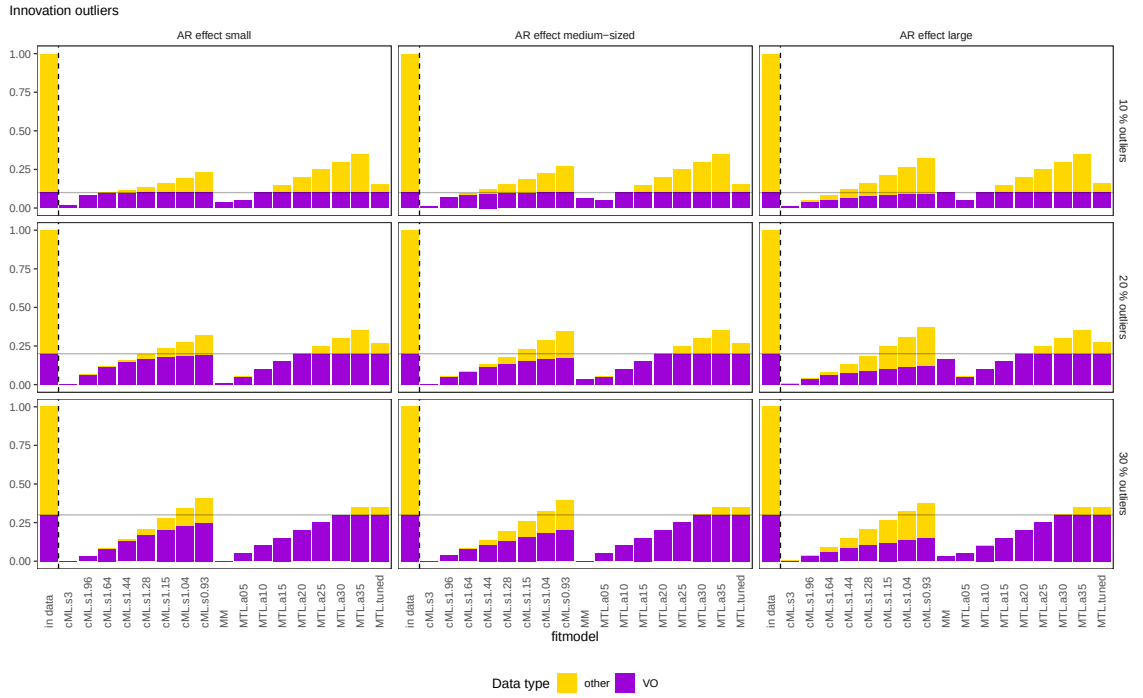
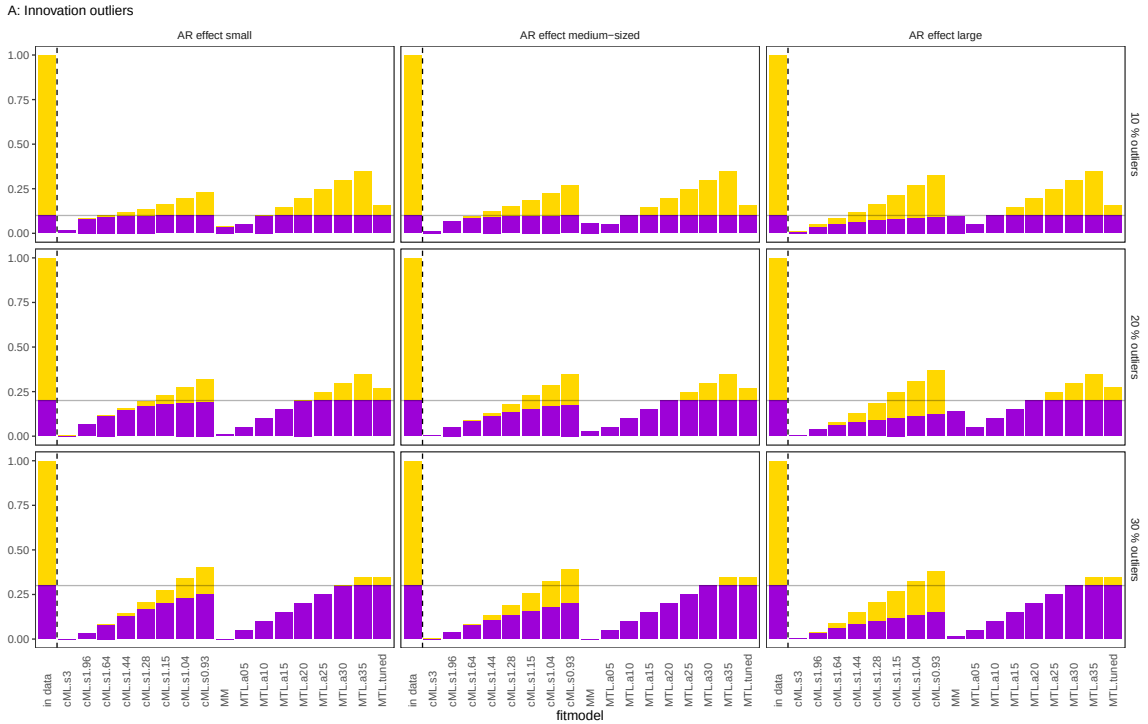


Figure 41

Relative frequencies of outliers excluded for data with no measurement error for long time series ($T = 800$) with randomly occurring outliers, and models fitted without measurement error.



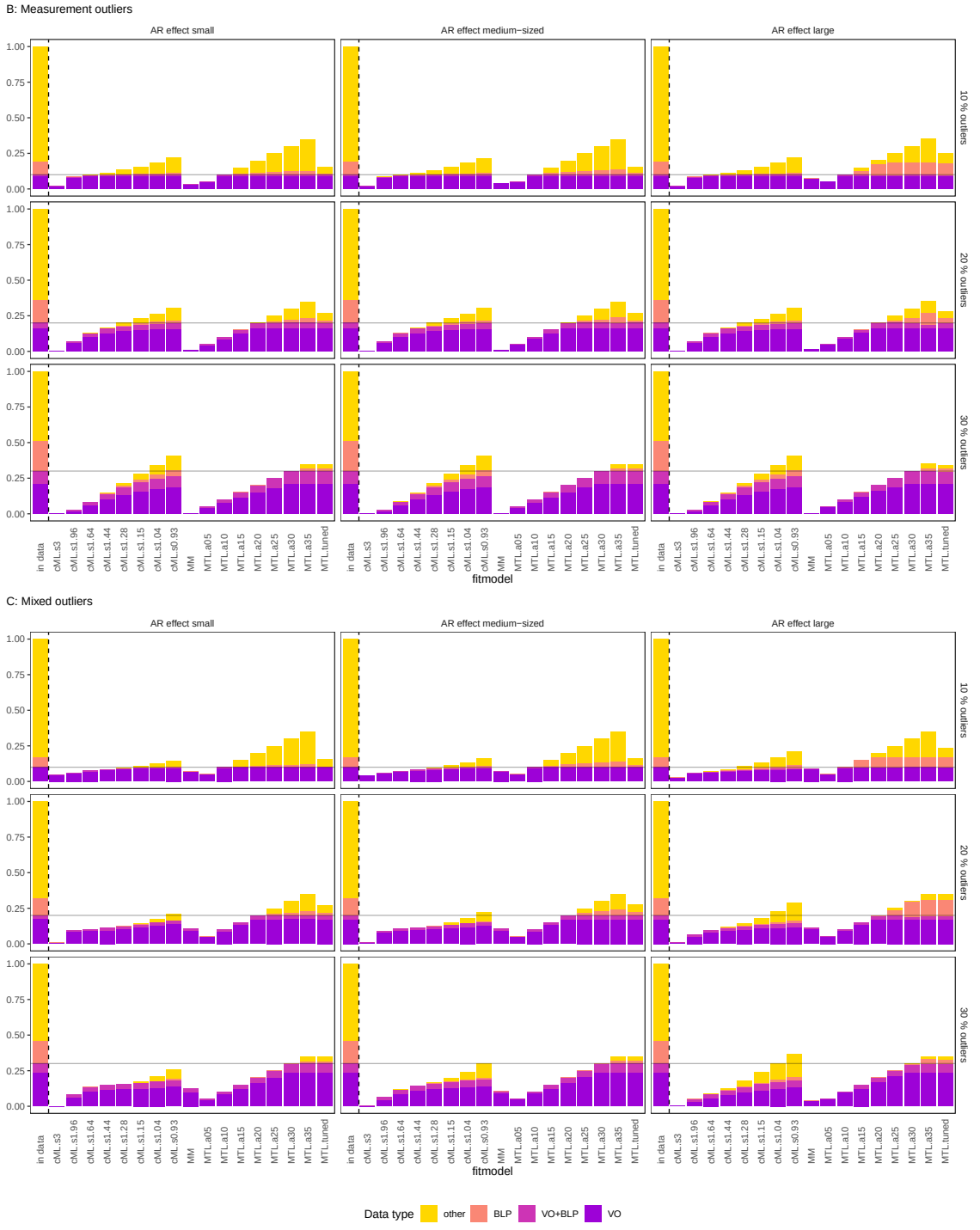


Figure 42

Relative frequencies of outliers excluded for data with little measurement error for long time series ($T = 800$) with randomly occurring outliers, and models fitted without measurement error.

2.2.5 Temporally dependent outliers

The figures in this section display performance of the different robust methods under different types of outliers that occur with varying temporal dependence. The section is structured as section 2.2.4. That is, we again cast model performance in terms of the MSE of the AR parameter, the MSPE of the overall model and in terms of how a method handles outliers during estimation. Figures 43, 48 and 53 show the estimation error of the AR effect for data with no measurement error and short, medium-length and long time series, respectively, and Figures 44, 49 and 54 show the estimation error of the AR effect for short, medium-length and long time series and data with measurement error. Figures 45, 50 and 55 show the prediction error for data with no measurement error and Figures 46, 51 and 56 show the prediction error for data with measurement error, and Figures 47, 52 and 57 the outlier handling for short, medium-length and long time series, where outliers occurred with high temporal dependence. We limit the presentation to models fitted without measurement error. Results on model performance given modeled measurement error are shown in the following section.

For the MSE- and MSPE-figures, we again show the performance measure (y-axis) as a function of the (robust) modeling approach used (x-axis). The different degrees of temporal dependence in the outliers are distinguished by point shape and bar color: Diamonds and colored bars mark randomly occurring outliers, triangles and gray-scaled bars mark occurrence with medium temporal dependence and circles and gray-scaled bars occurrence with high temporal dependence. The winning models are still marked for randomly occurring outliers. The case of mixed outliers is not included since those outliers could only occur randomly. The outlier handling figures again show the relative frequencies of vertical outliers (VO), bad leverage points (BP), simultaneous vertical outlier and bad leverage points (VO+BLP) and other data points (y-axis) in the actual data and excluded by each robust method (x-axis). With the high temporal dependence condition we focus on a scenario that maximally contrasts with the random occurrence condition.

Looking at the MSE-figures, it seems that, in case of innovation outliers, higher outlier dependency leads to higher MSEs. This could reflect the increased probability that good leverage points are more likely to be contaminated with new innovation and hence vertical outliers, which would likely impair estimation accuracy. In case of measurement outliers, however, we see that the effects of outlier dependence reverse for medium-sized and large population-level AR effects. Here higher temporal dependence seems beneficial. This is surprising on first sight, but probably again due to the impact of the different data point types produced: Measurement outliers occurring with a high temporal dependence cause a substantial number of data points that are simultaneous vertical outliers and bad leverage points. Since the outlier effect is designed to be large but not extreme in our simulation study, it is likely that, for the estimation of large(r) AR effects, these data points function more as good leverage points than as vertical outliers or bad leverage points. In fact, we already saw this happen in the illustrative example in manuscript. Taking a look at the outlier handling figures, we indeed see that (a) there are higher proportions of simultaneous vertical outliers and bad leverage points in the data, and (b) less of these are handled as compared to when outliers occur randomly.

Looking at the MSPE-figures it seems that higher outlier dependence generally has positive effects on overall predictive performance, also for innovation outliers. This discrepancy is probably partly due to the other parameter estimators of the AR model behaving differently under varying outlier dependencies than the AR parameter estimator.

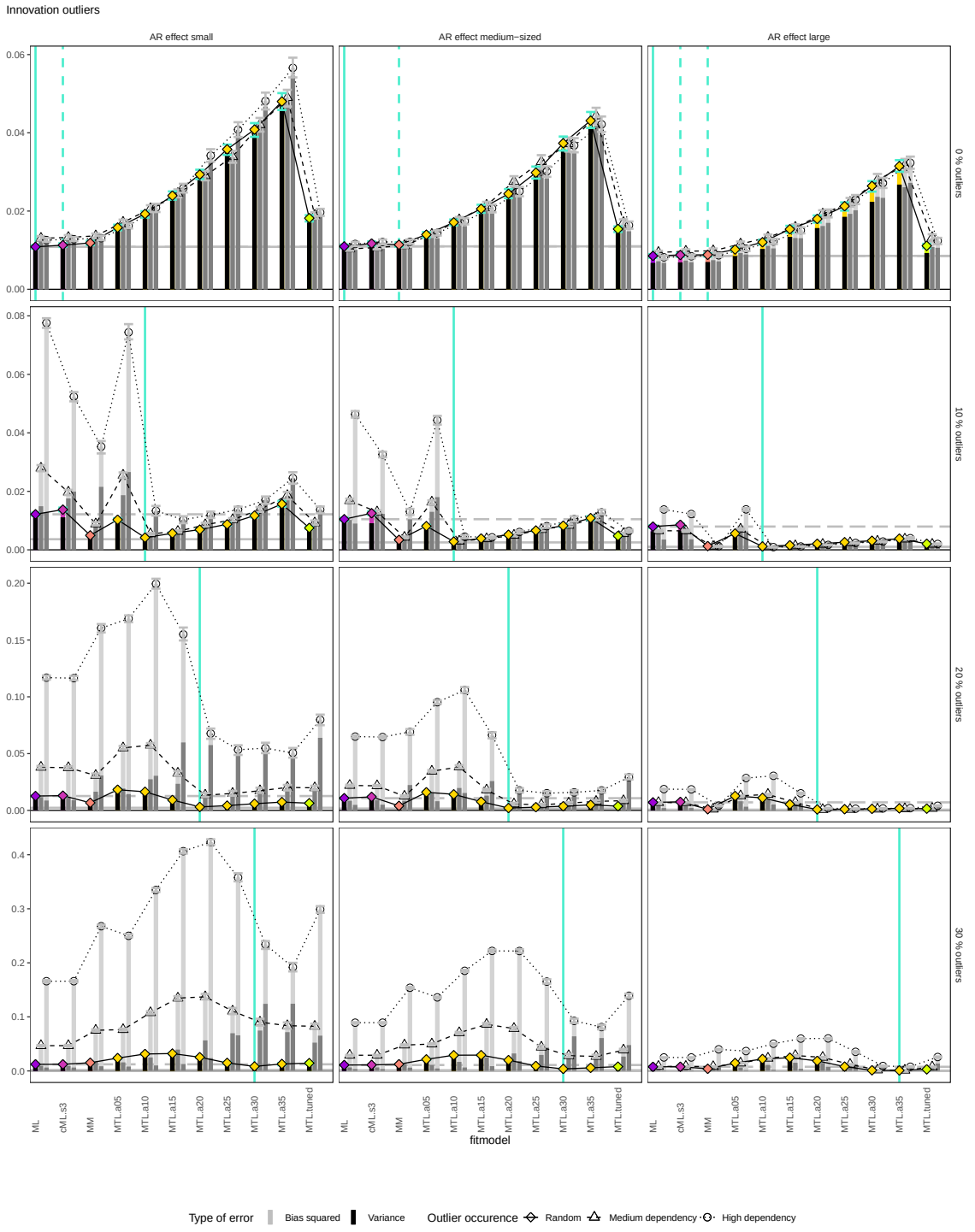


Figure 43

Estimation error AR effect for data with no measurement error and short time series ($T = 80$).

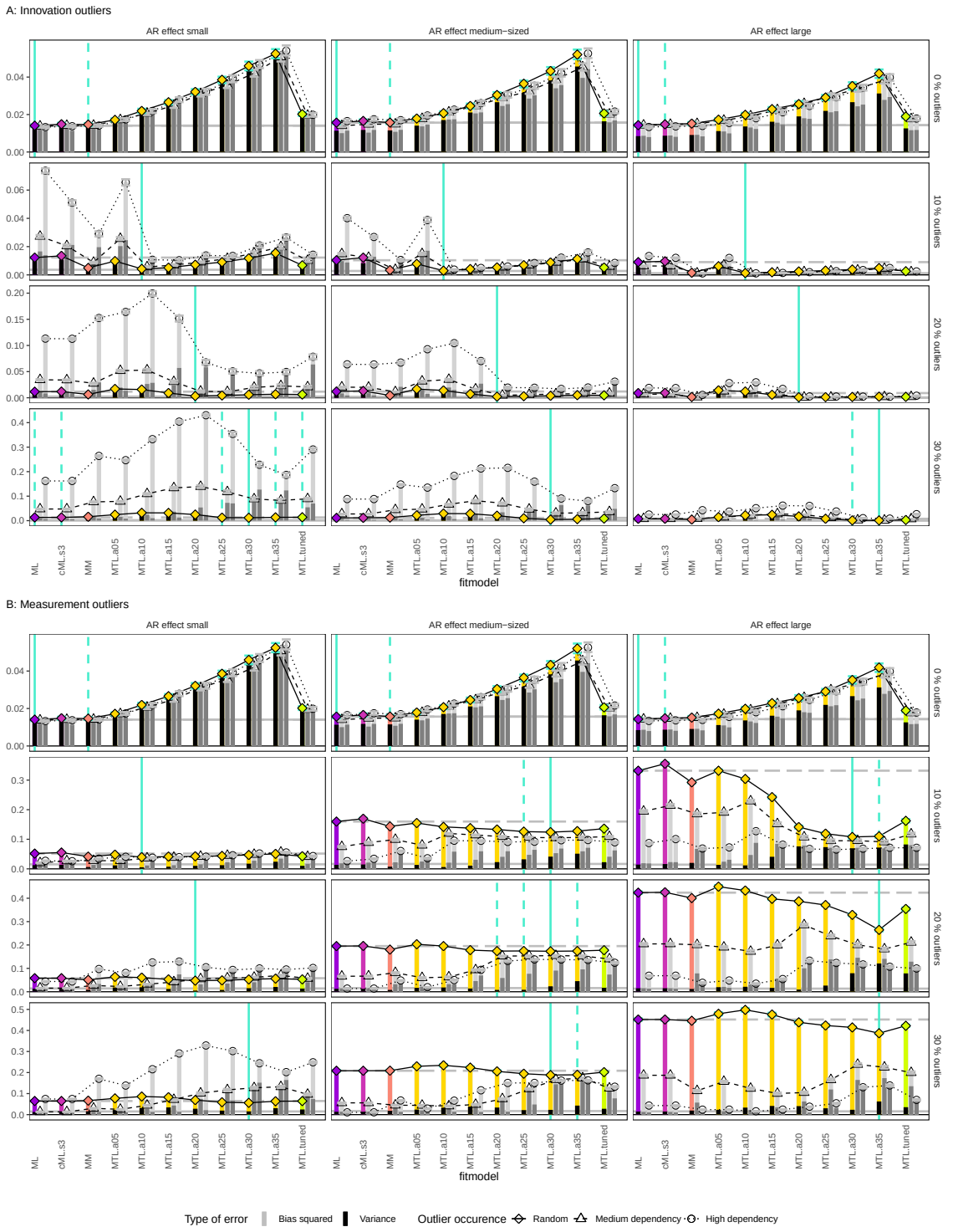


Figure 44

Estimation error AR effect for data with little measurement error and short time series ($T = 80$).

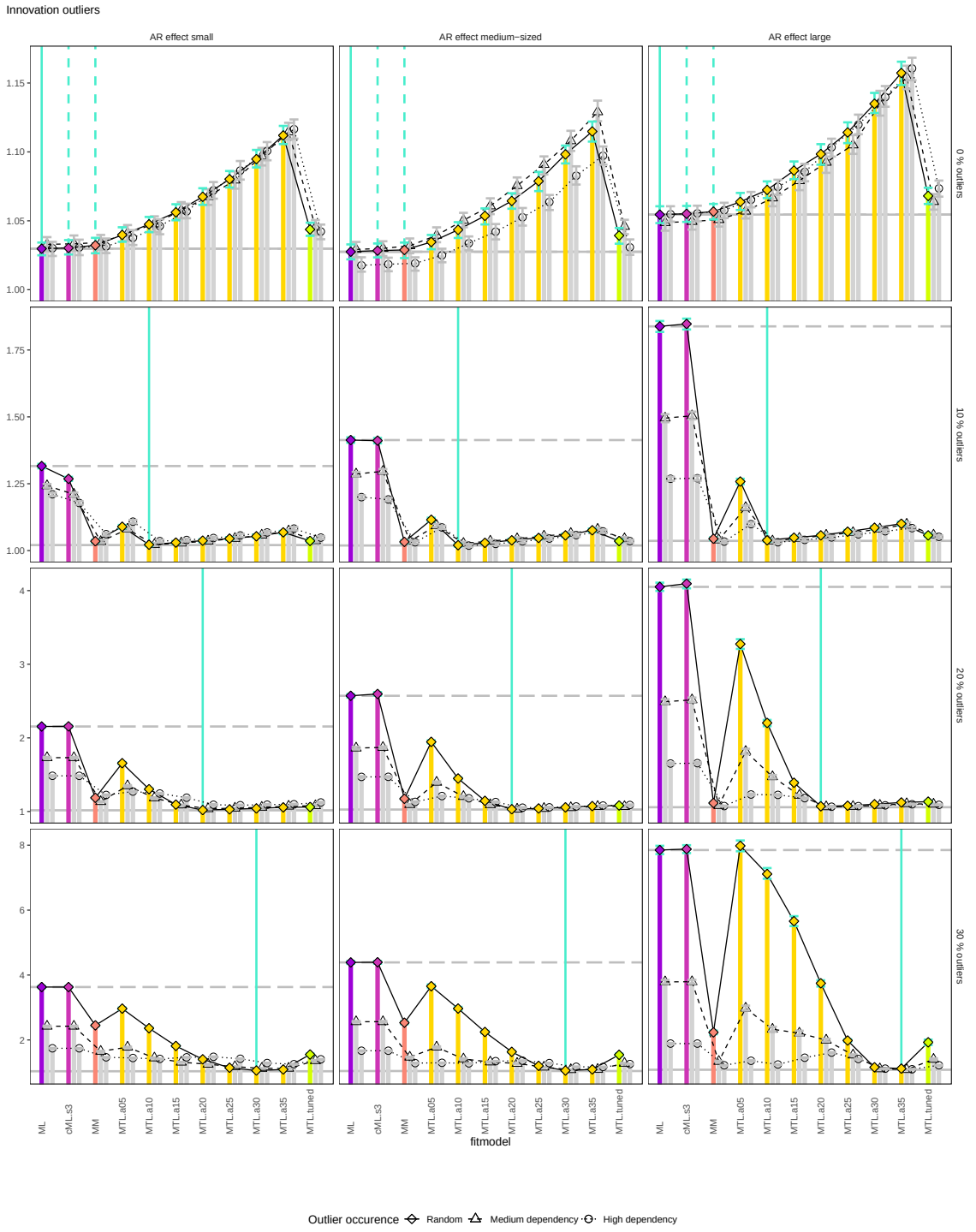


Figure 45

Prediction error for data with no measurement error and short time series ($T = 80$).

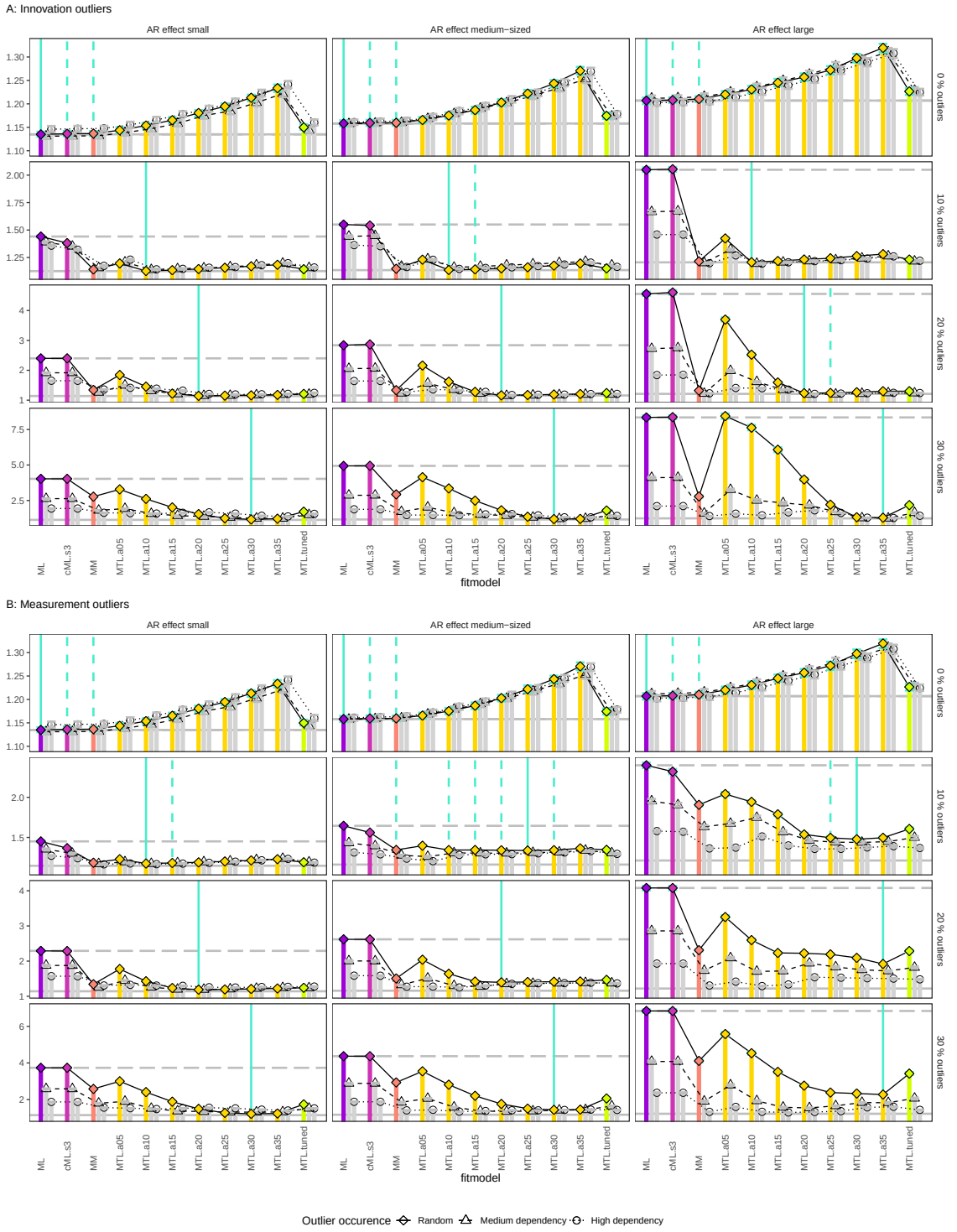


Figure 46

Prediction error for data with little measurement error and short time series ($T = 80$).

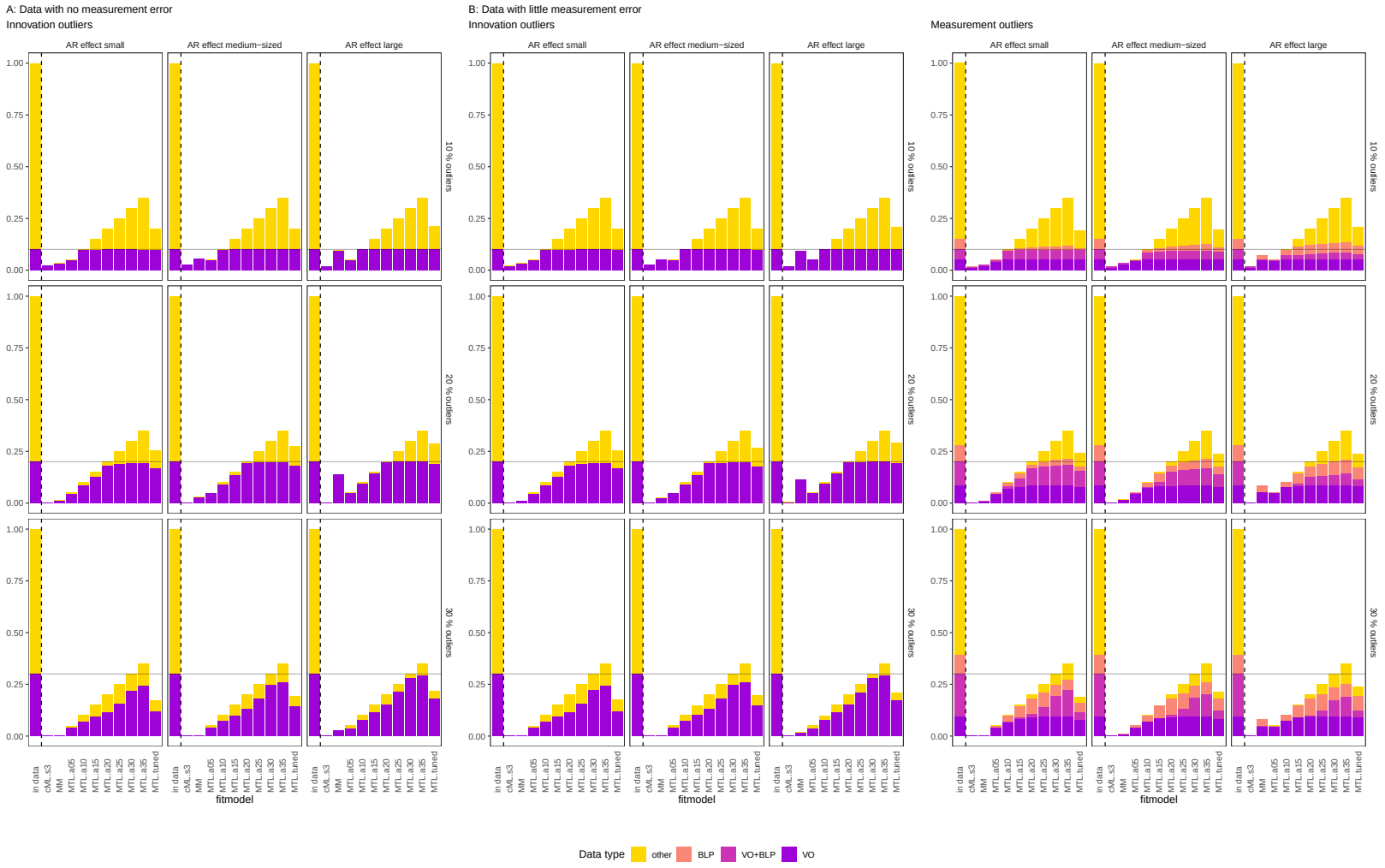


Figure 47

Relative frequencies of outliers excluded for short time series ($T = 80$) with outliers occurring with high temporal dependency, and models fitted without measurement error.

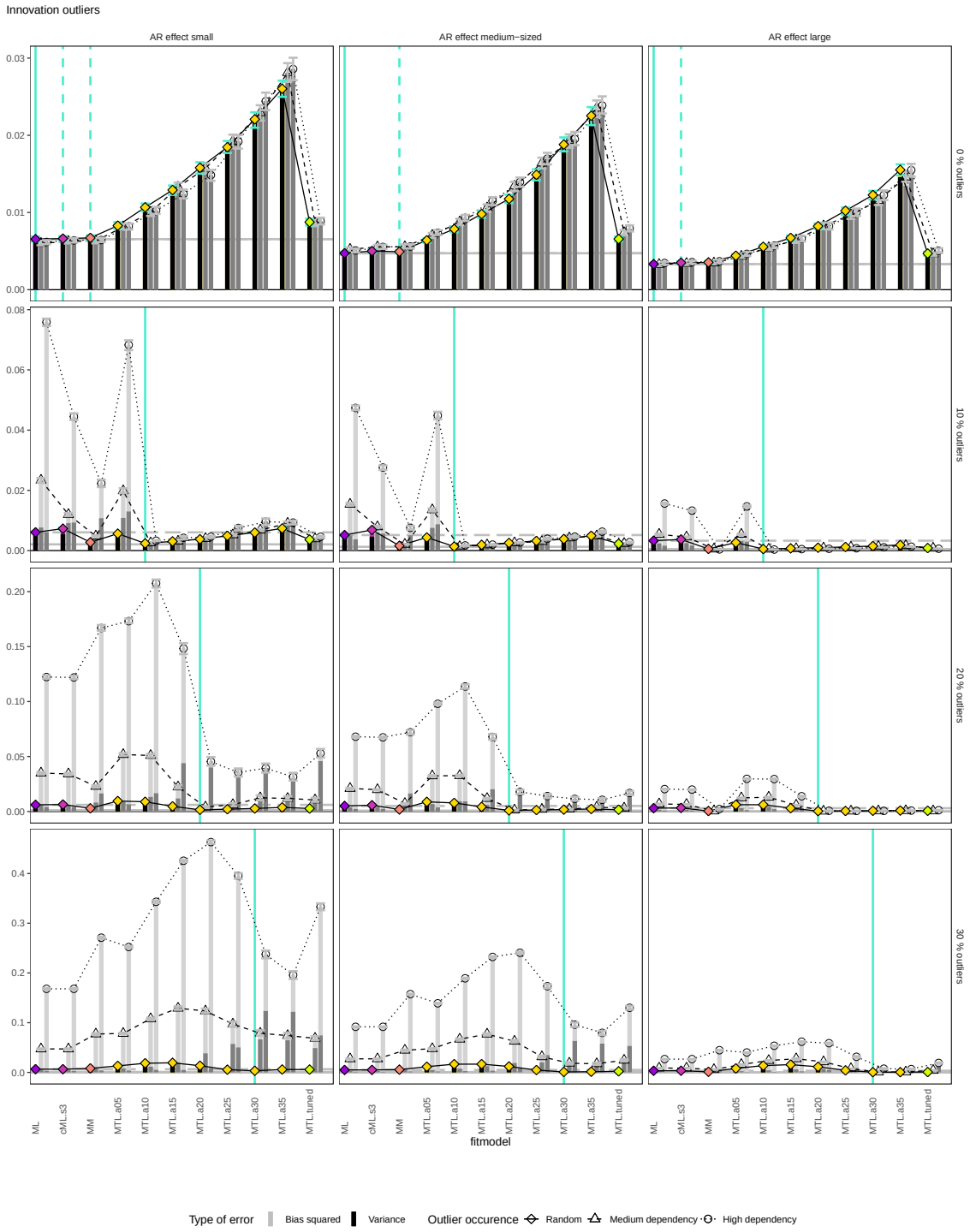


Figure 48

Estimation error AR effect for data with no measurement error and medium-length time series ($T = 160$).

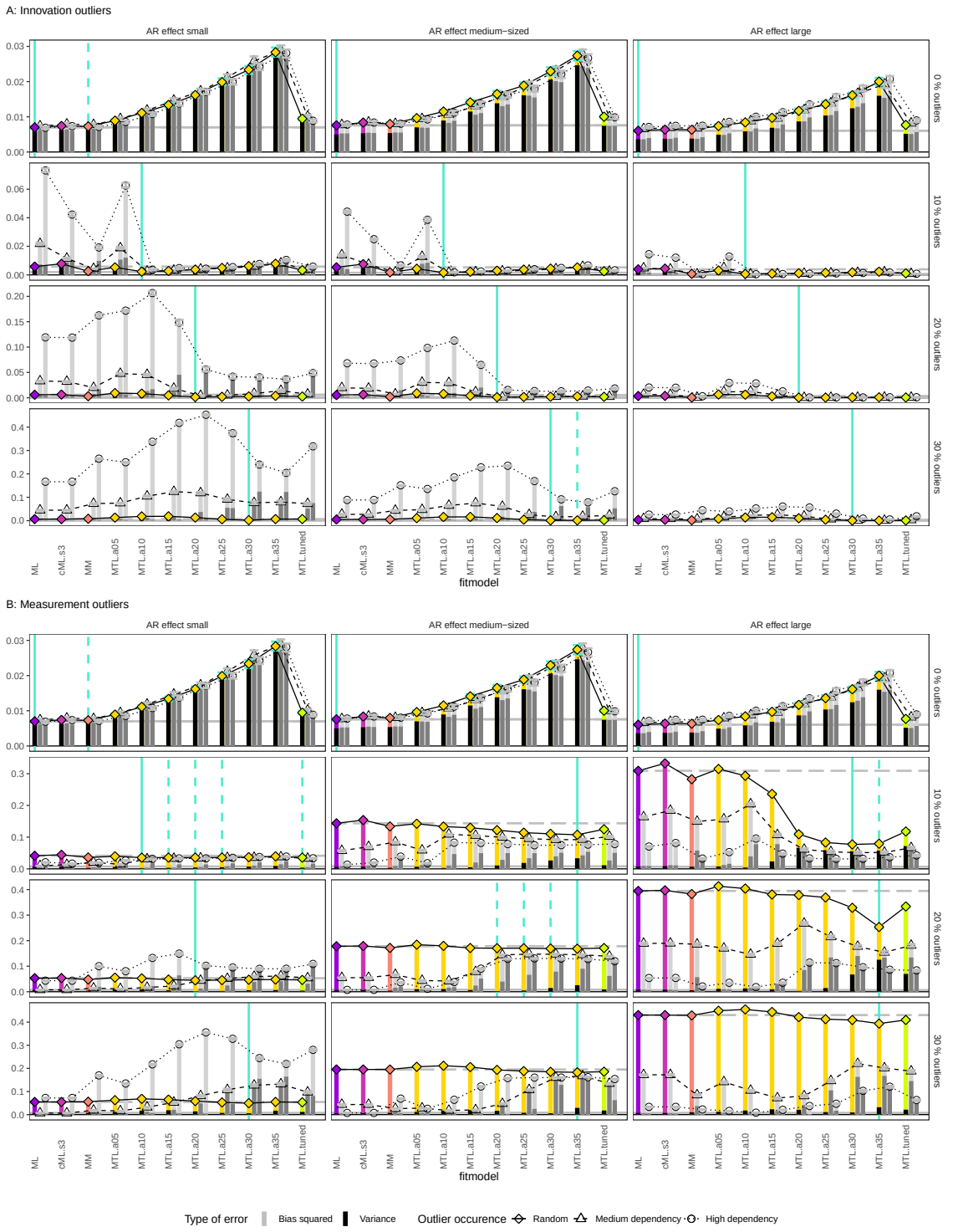


Figure 49

Estimation error AR effect for data with little measurement error and medium-length time series ($T = 160$).

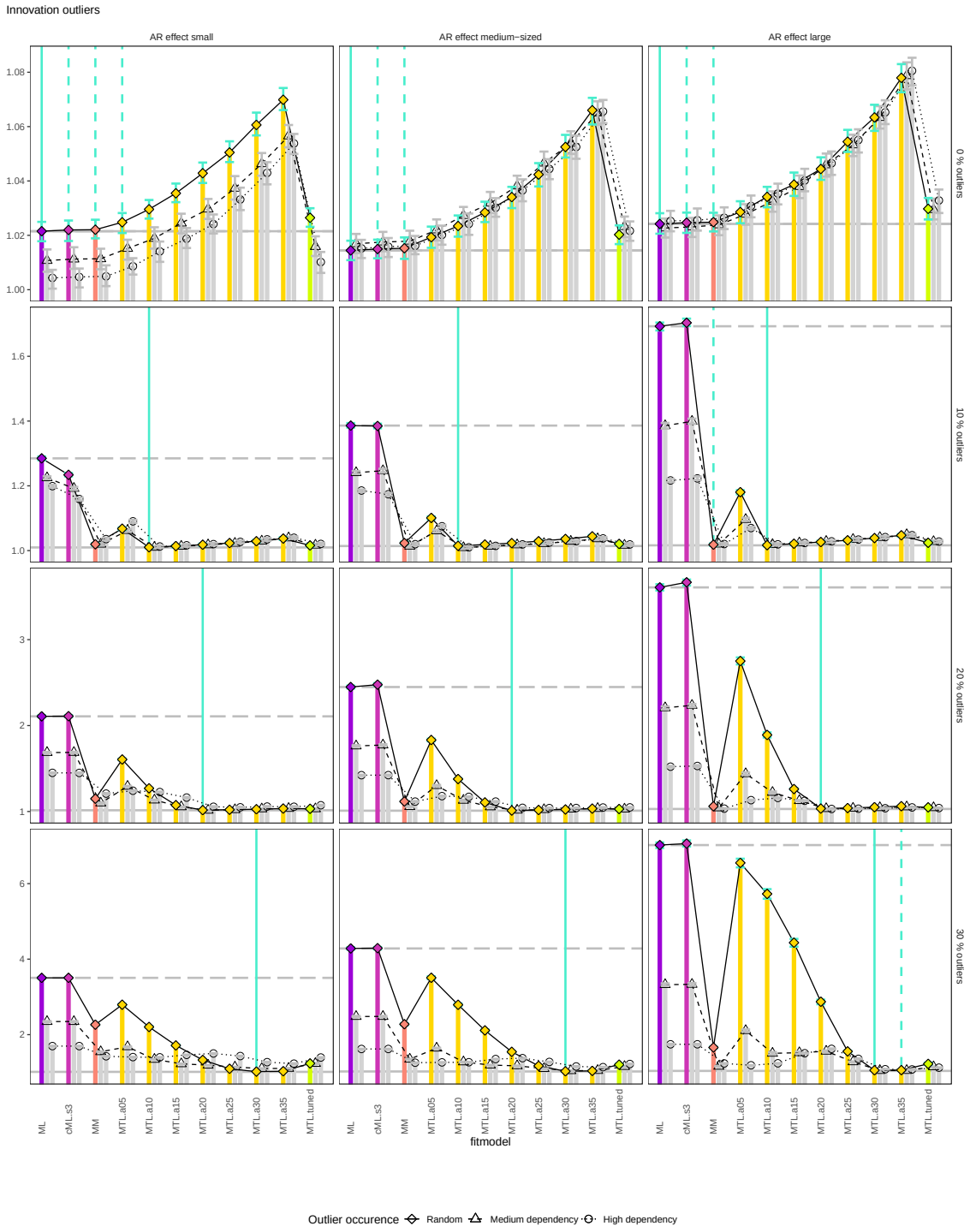


Figure 50

Prediction error for data with no measurement error and medium-length time series ($T = 160$).

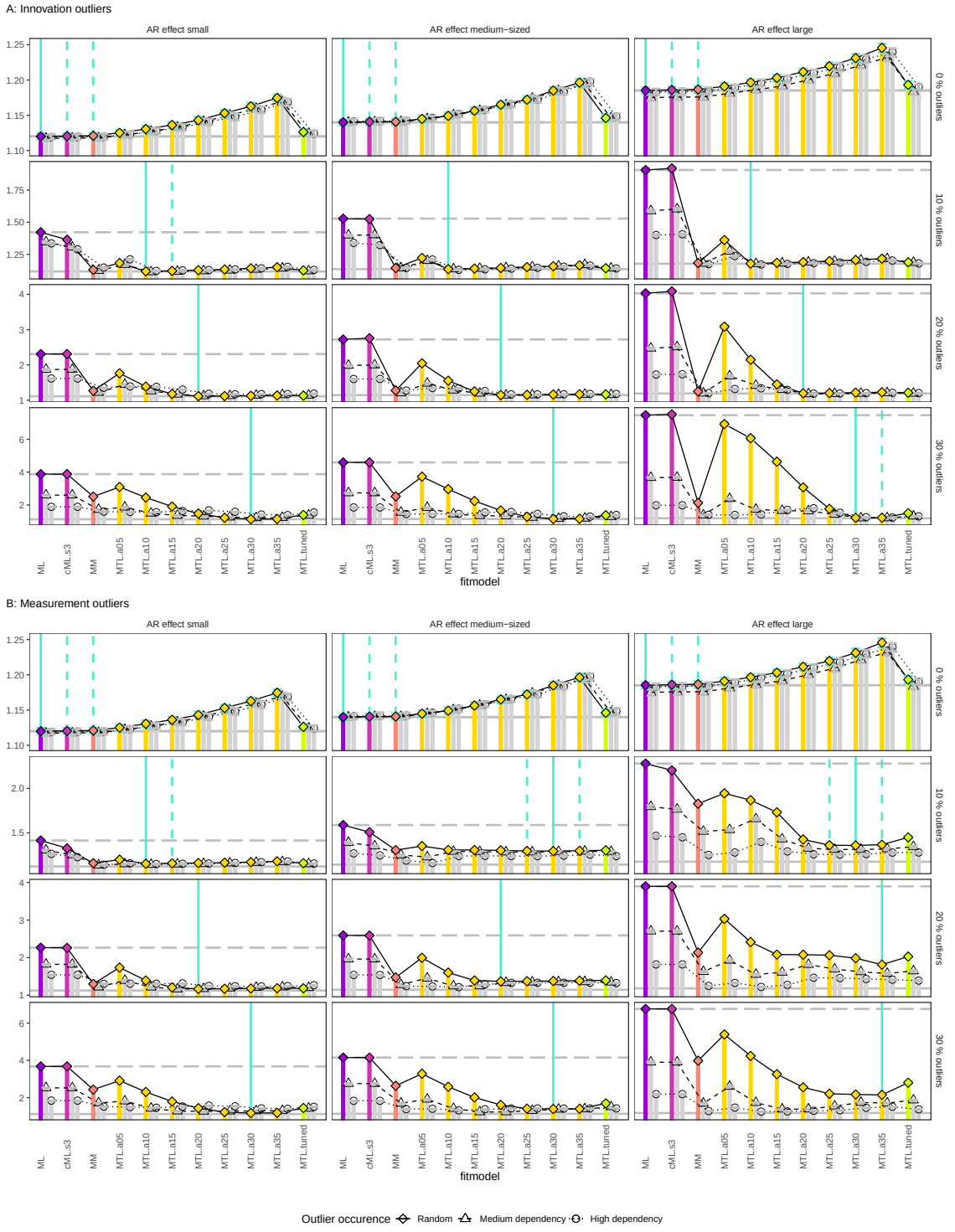


Figure 51

Prediction error for data with little measurement error and medium-length time series ($T = 160$).

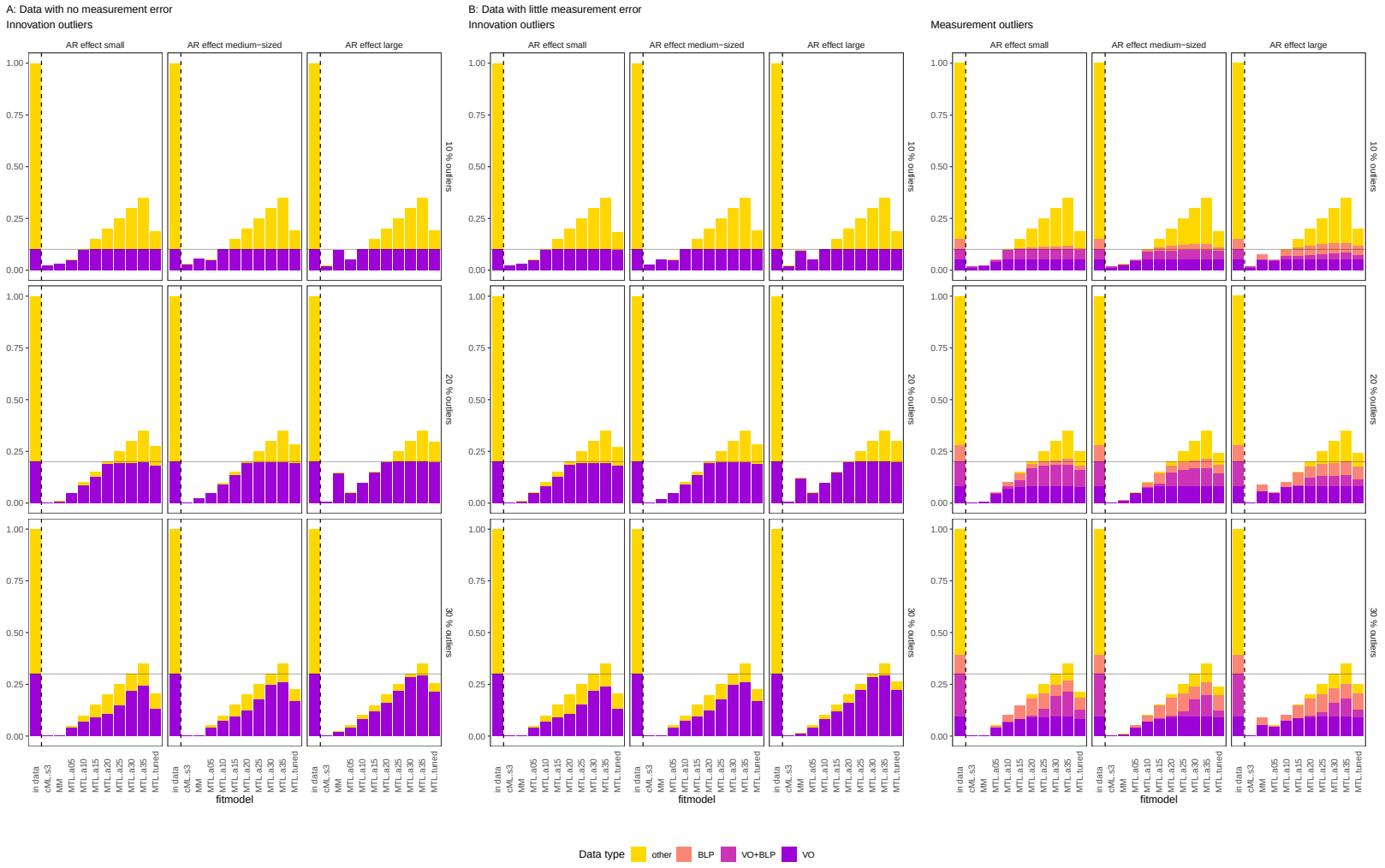


Figure 52

Relative frequencies of outliers excluded for medium-length time series ($T = 160$) with outliers occurring with high temporal dependency, and models fitted without measurement error.

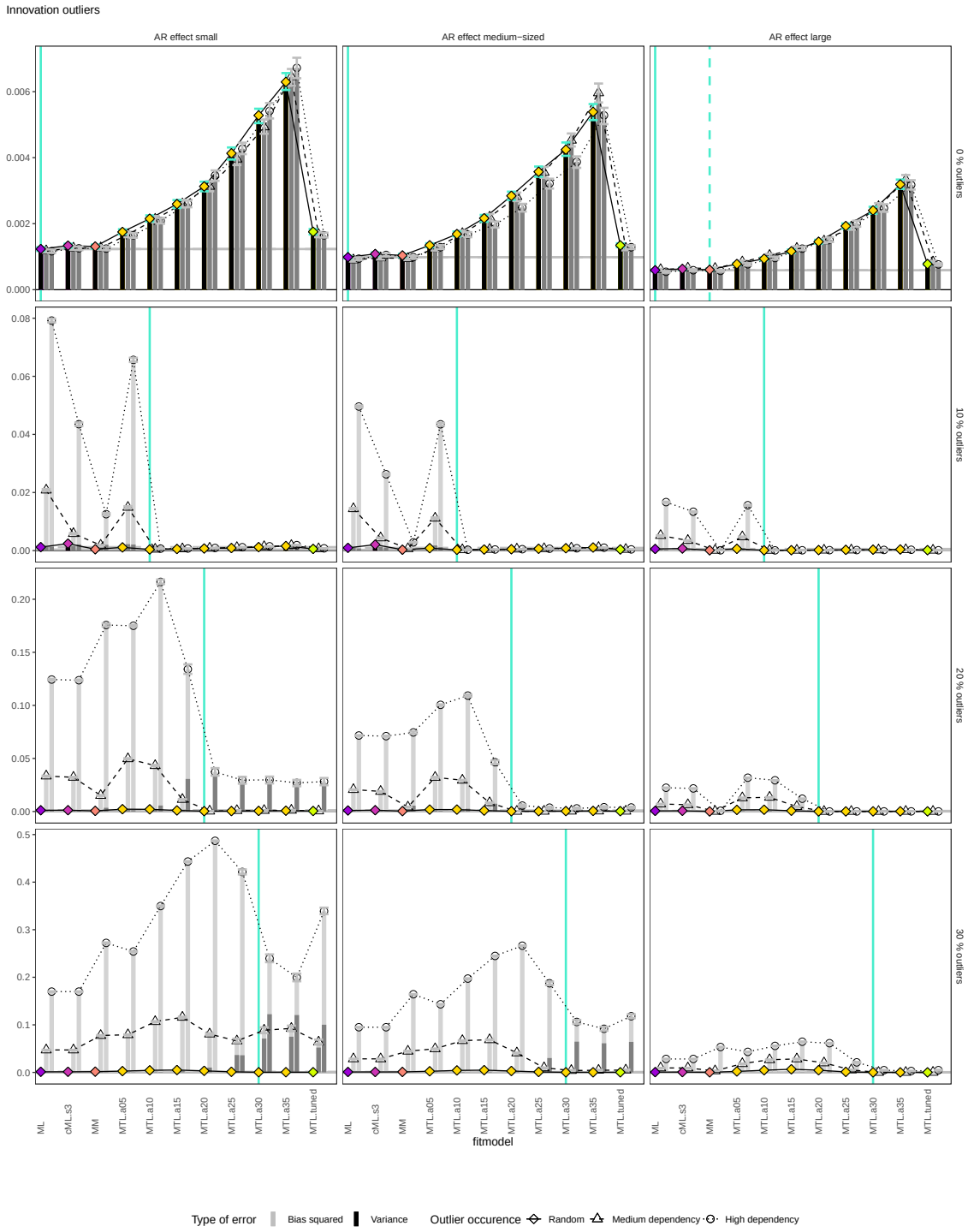


Figure 53

Estimation error AR effect for data with no measurement error and long time series ($T = 800$).

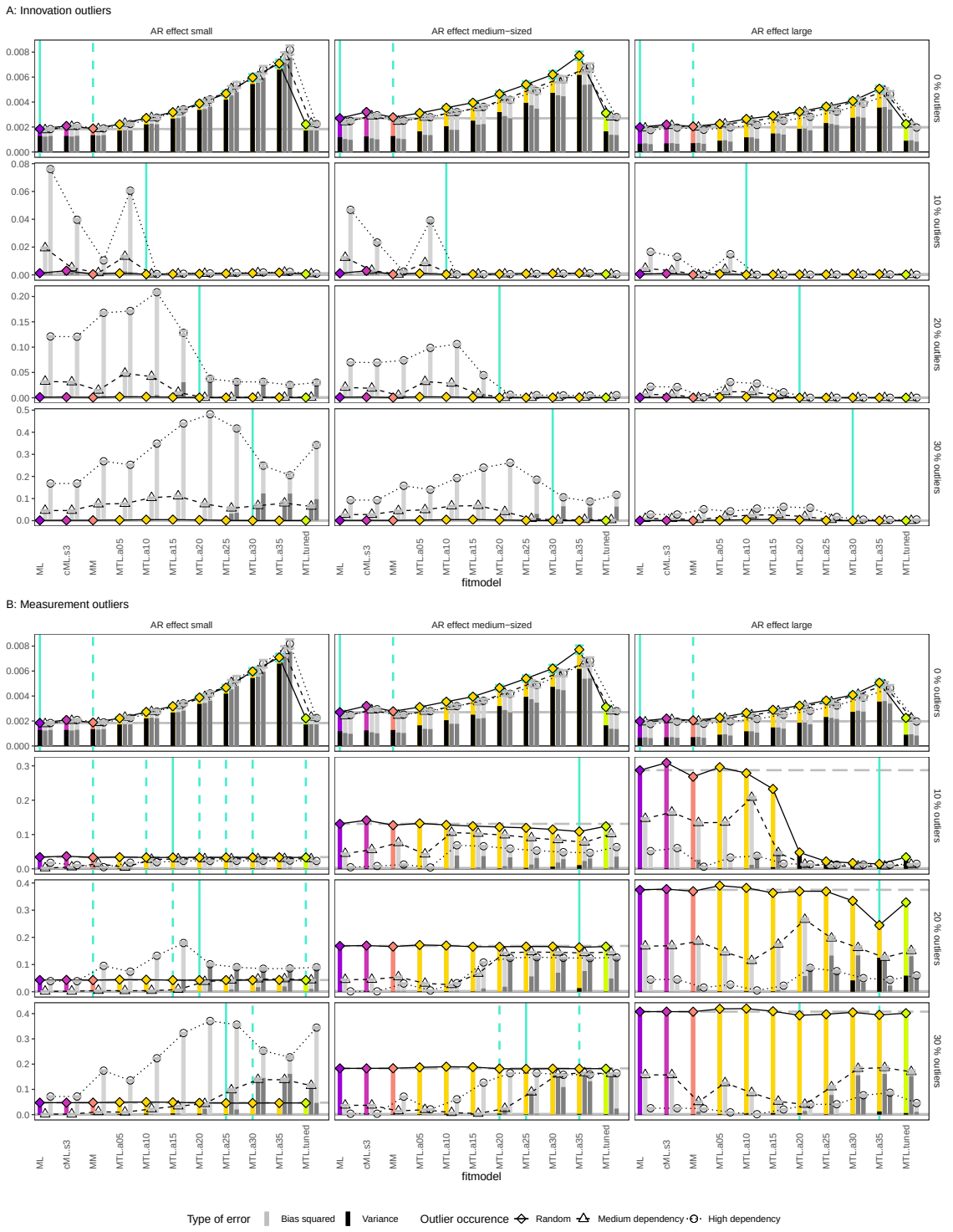


Figure 54

Estimation error AR effect for data with little measurement error and long time series ($T = 800$).

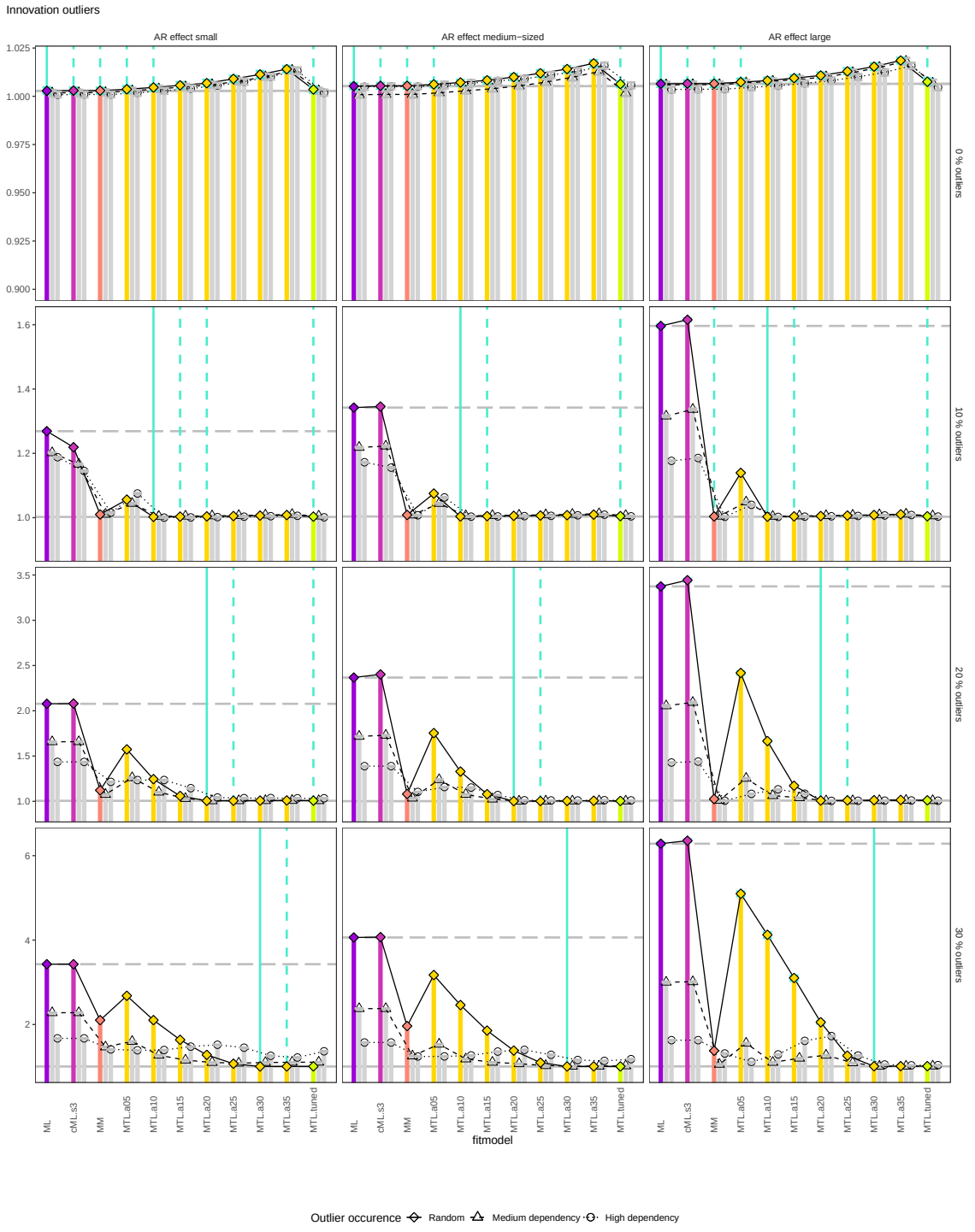


Figure 55

Prediction error for data with no measurement error and long time series ($T = 800$).

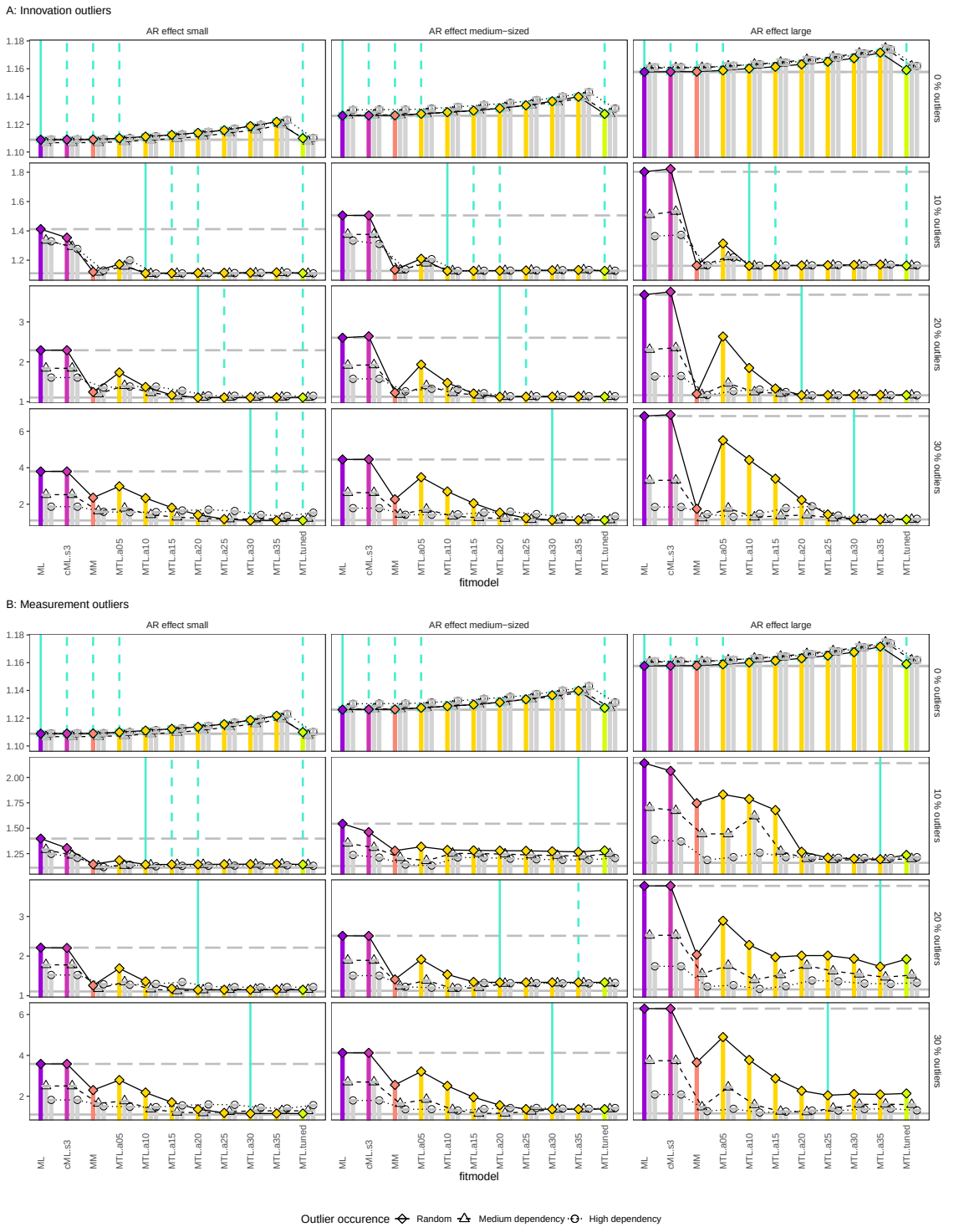


Figure 56

Prediction error for data with little measurement error and long time series ($T = 800$).

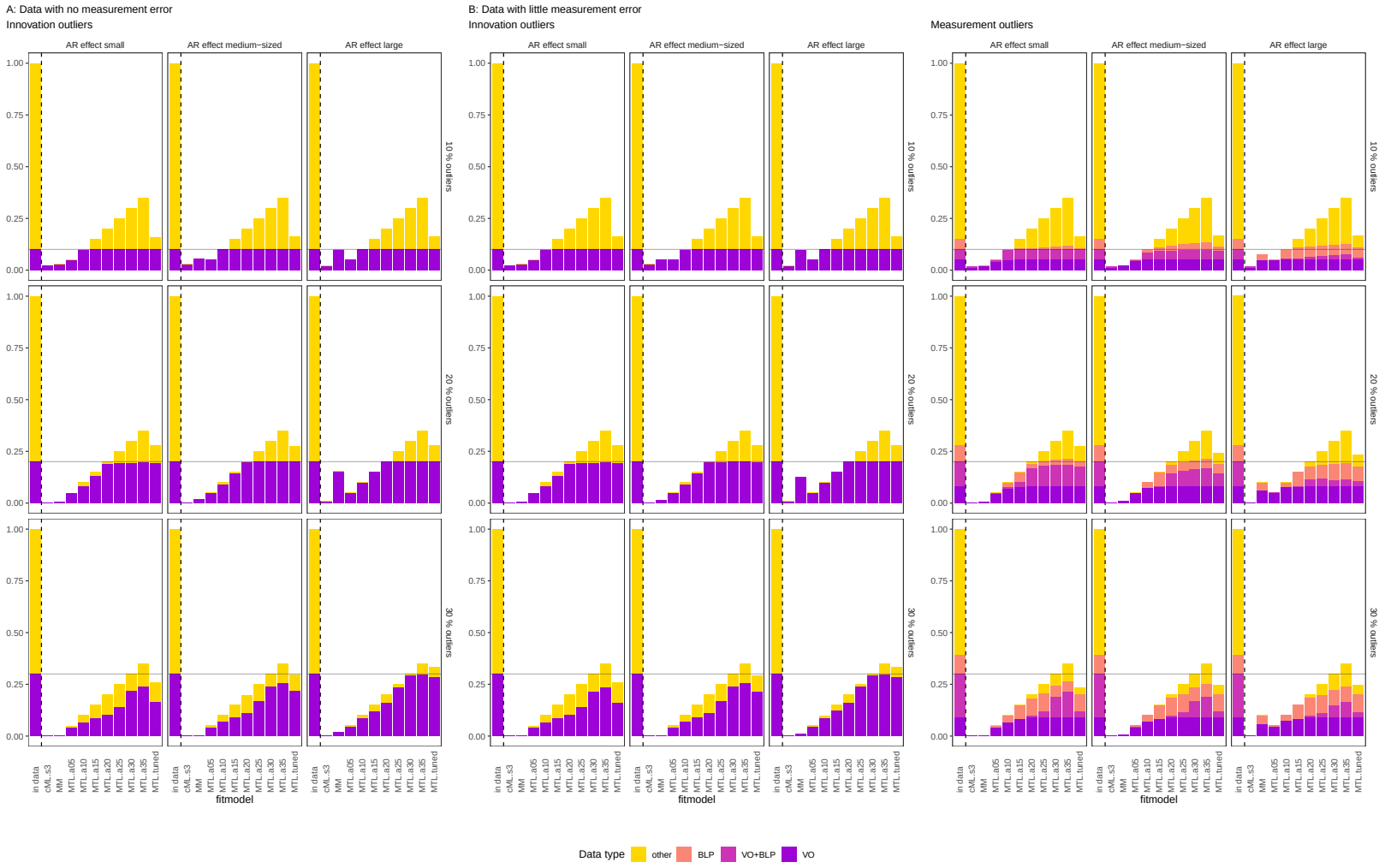


Figure 57

Relative frequencies of outliers excluded for long time series ($T = 800$) with outliers occurring with high temporal dependency, and models fitted without measurement error.

2.2.6 Modeling measurement error

The figures in this section display the effects of modeling measurement error on model performance. Figures 58, 60 and 62 show the estimation error of the AR effect and Figures 59, 61 and 63 the prediction error for short, medium-length and long time series, respectively. We limit the presentation to randomly occurring outliers. Results on model performance given temporally dependent outliers are shown in the previous section.

Per figure, the MSE or MSPE (y-axis) is again shown as a function of the estimation method (x-axis). Results pertain to measurement error-affected data with innovation outliers (Panel A), measurement outliers (Panel B) or mixed outliers (Panel C). Outlier proportions are distinguished by rows, population-level AR effect sizes by columns.
Estimation methods are reduced to classical ML and MTL but each method is implemented in a model without (diamonds) and with measurement error (triangles). Bars displaying the same estimator are grouped together, with colored bars referring to the models without measurement error and gray-scale bars to the models without. Bars are again decomposed into variance (either black or dark grey; lower bar part) and squared bias (color depending on method or light grey; upper bar part). Per estimator, the winning model variant with the lowest estimation error is marked in turquoise.

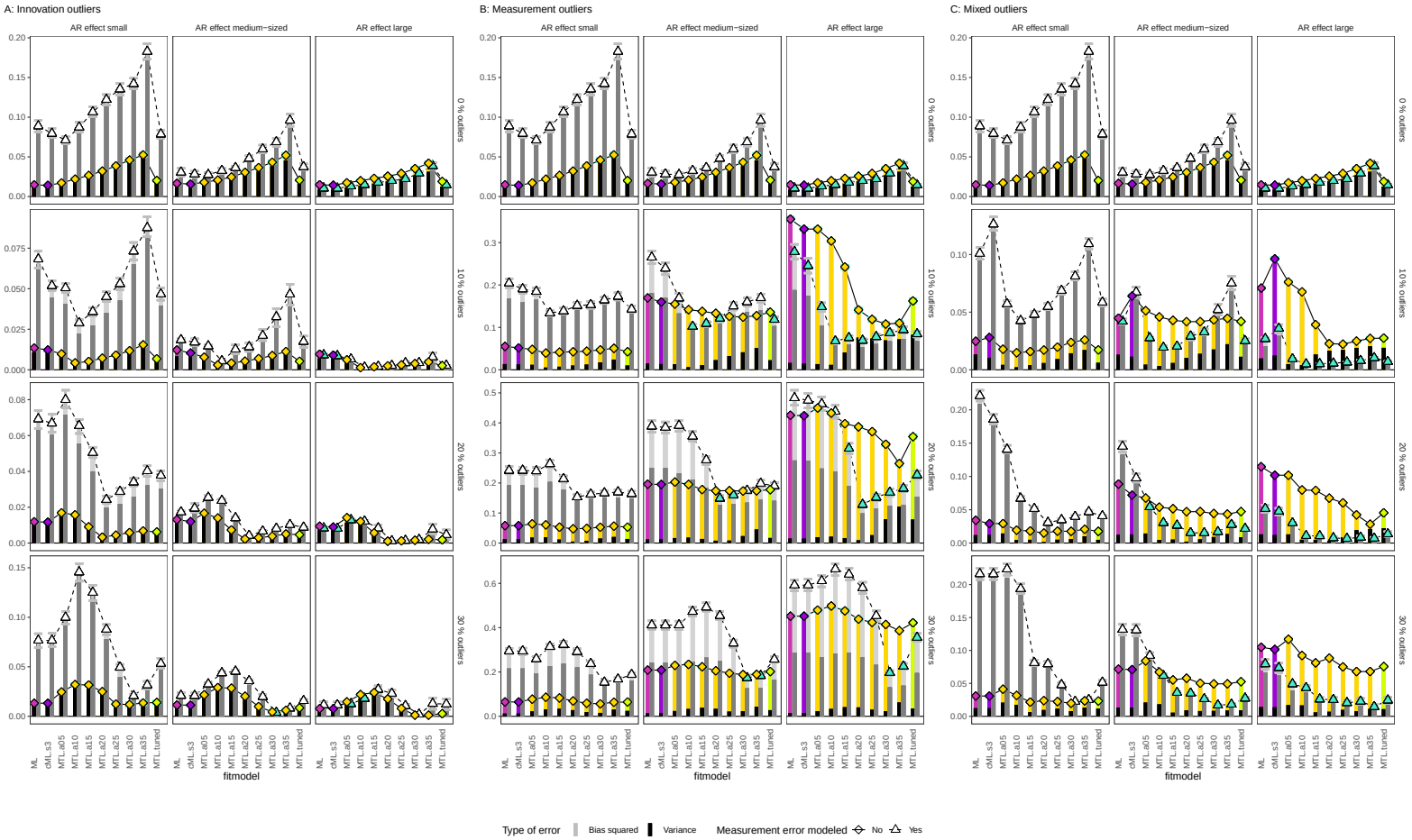


Figure 58

Estimation error AR effect for data with little measurement error for short time series ($T = 80$) with randomly occurring outliers.

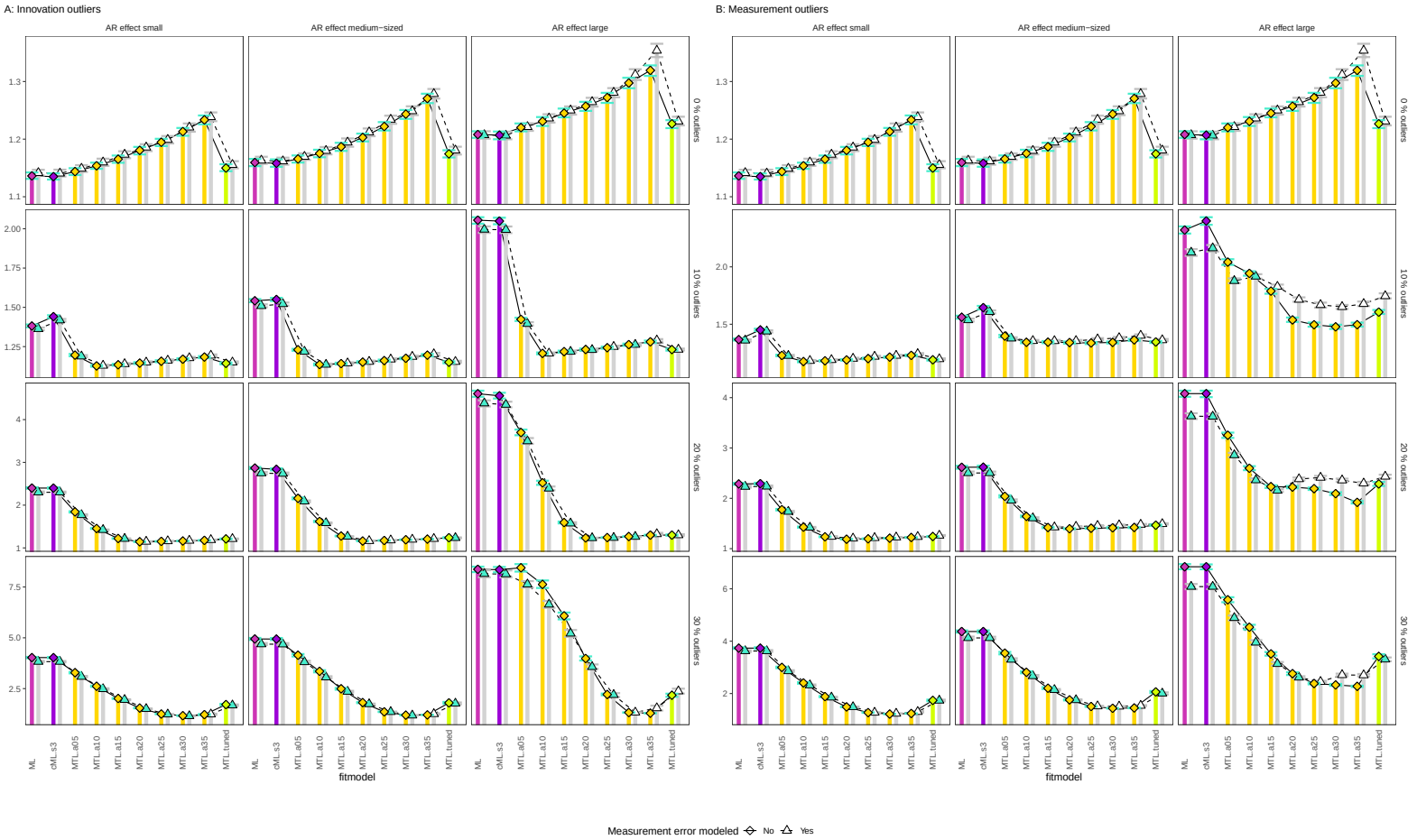


Figure 59

Prediction error for data with little measurement error for short time series ($T = 80$) with randomly occurring outliers.

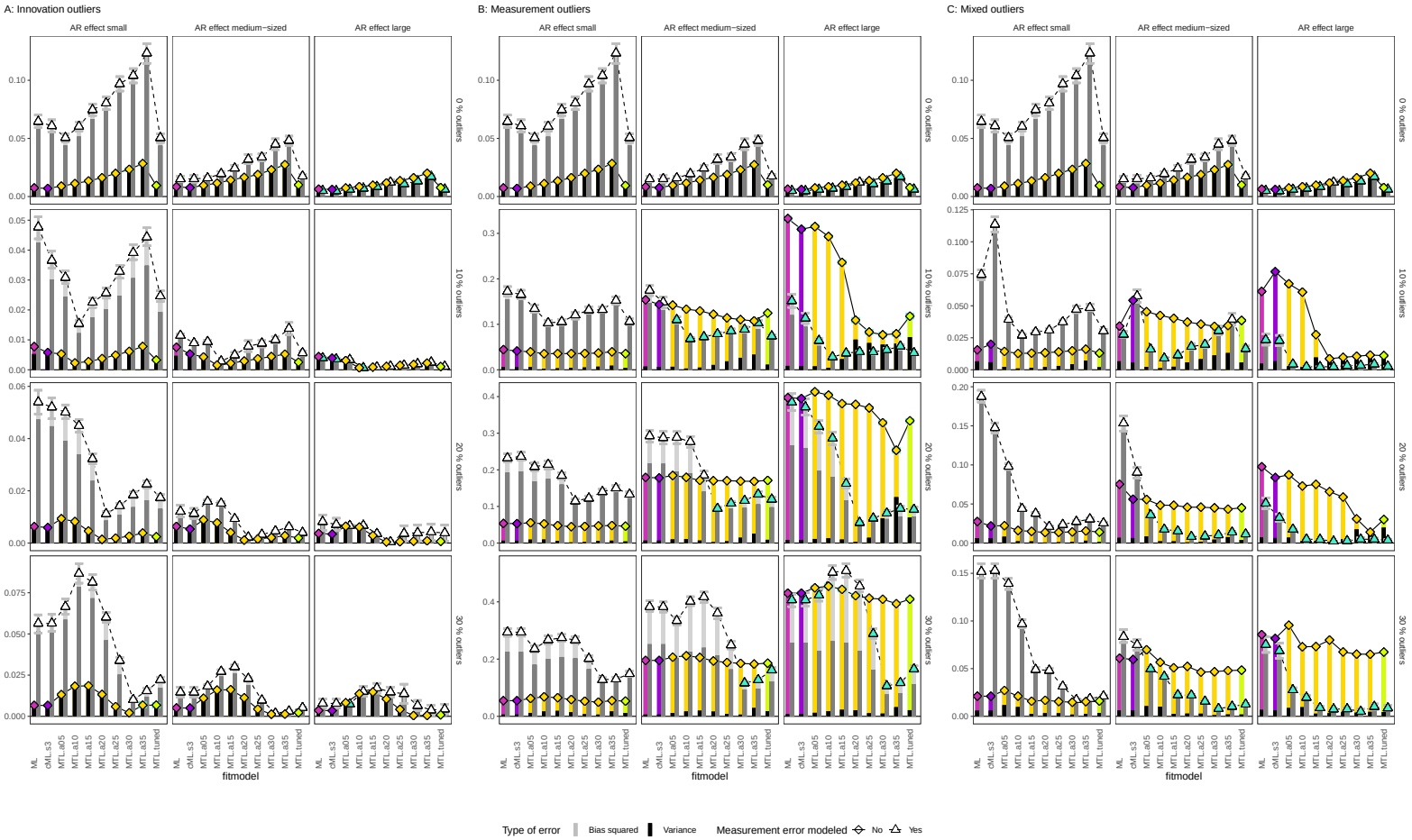


Figure 60

Estimation error AR effect for data with little measurement error for medium-length time series ($T = 160$) with randomly occurring outliers.

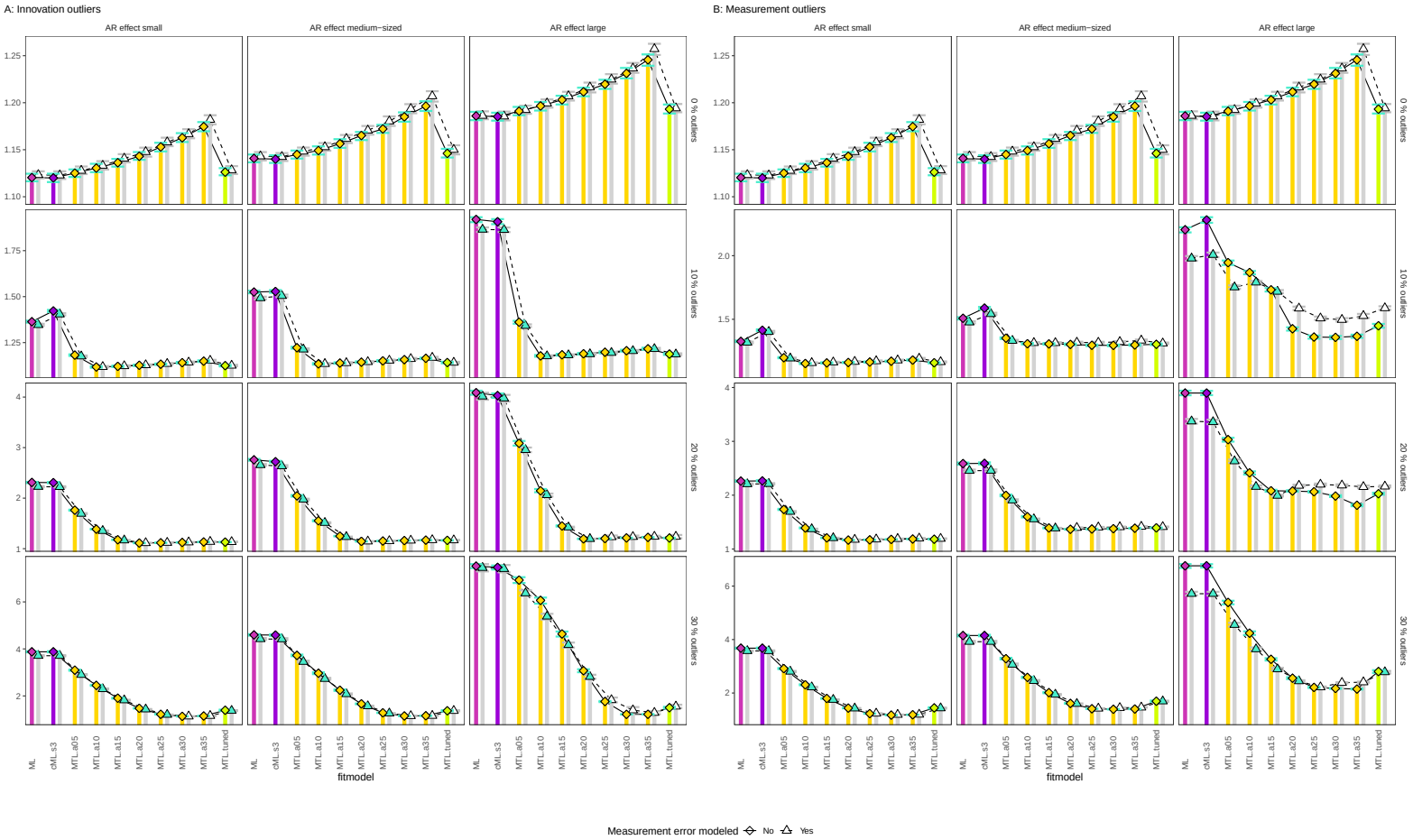


Figure 61

Prediction error for data with little measurement error for medium-length time series ($T = 160$) with randomly occurring outliers.

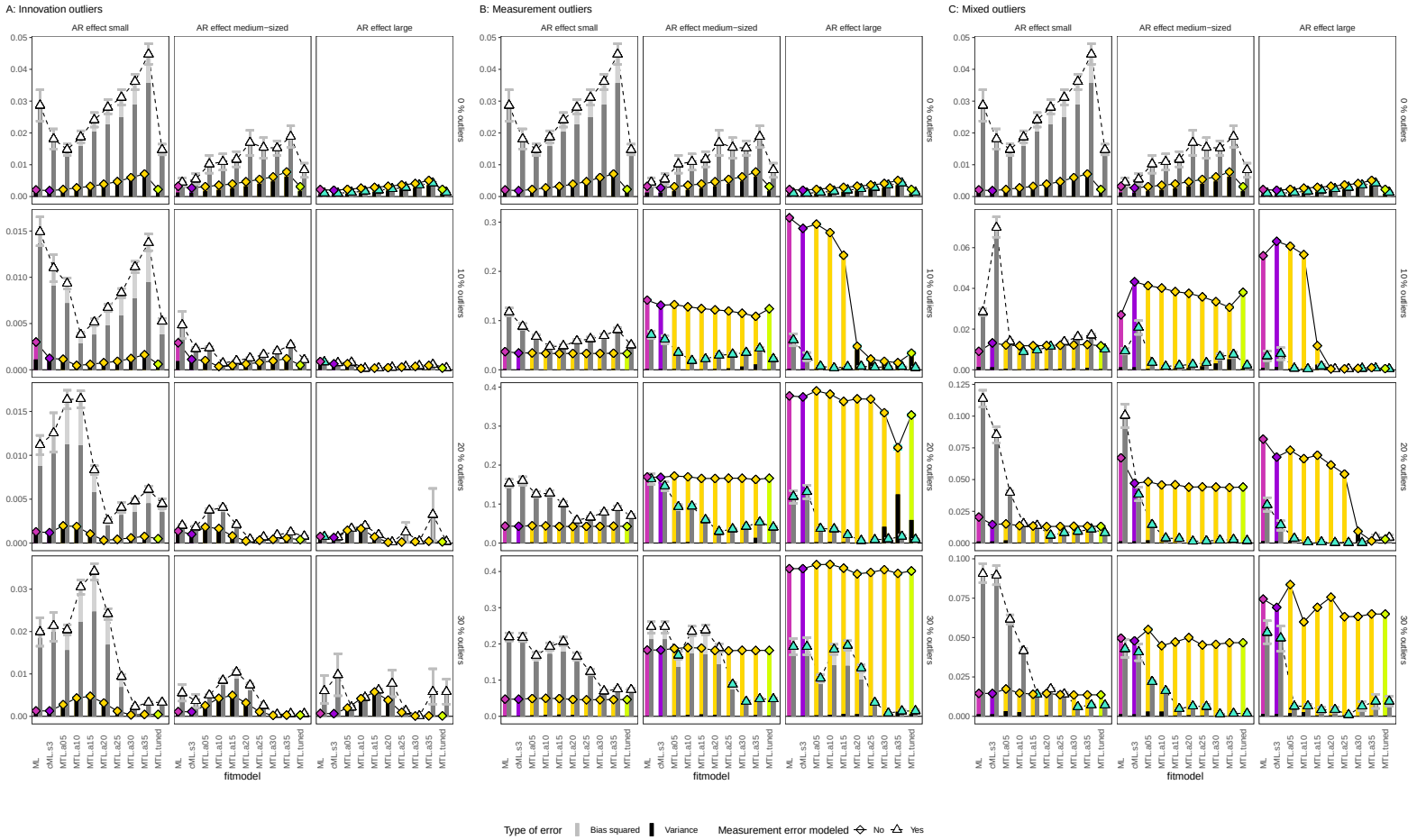


Figure 62

Estimation error AR effect for data with little measurement error for long time series ($T = 800$) with randomly occurring outliers.

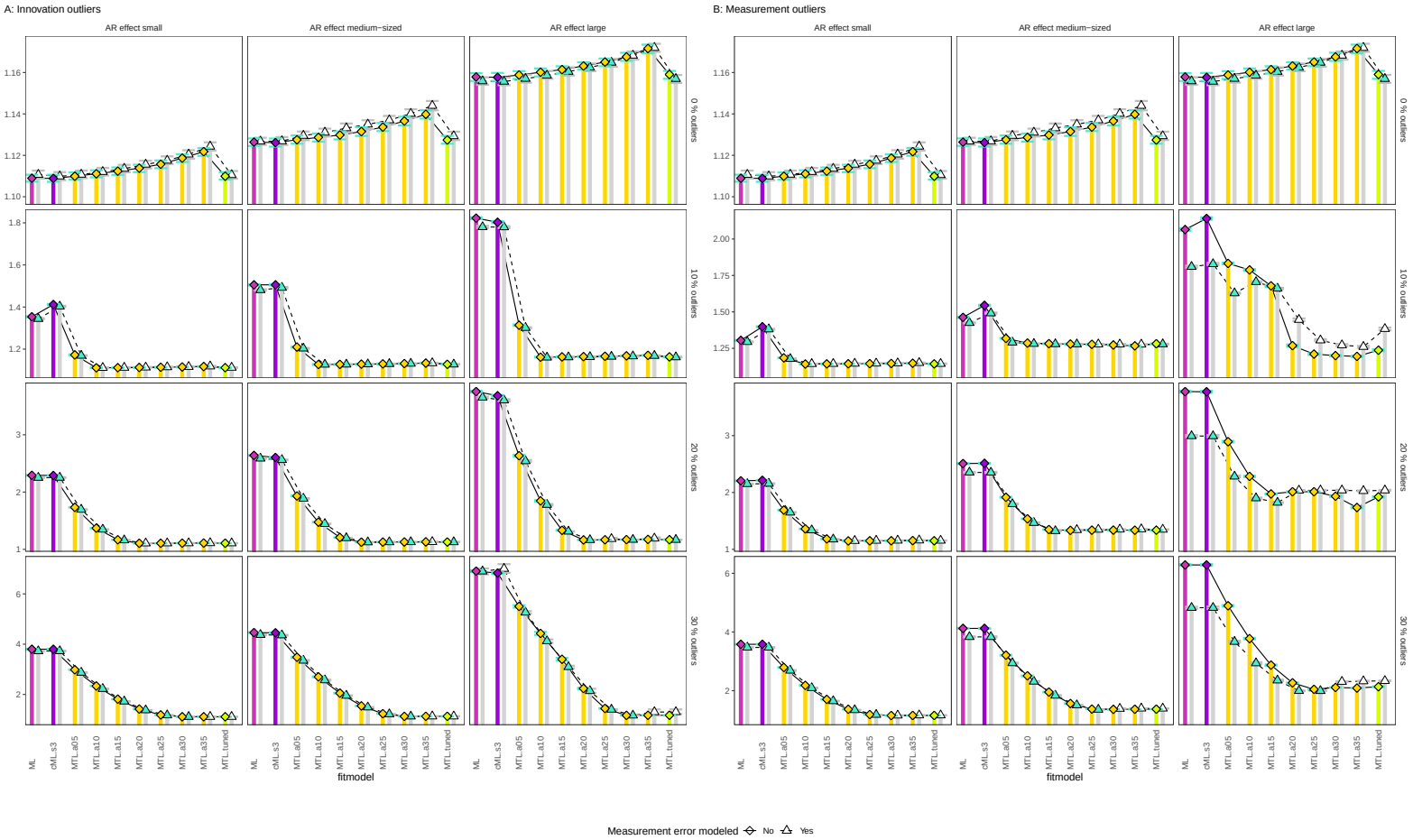


Figure 63

Prediction error for data with little measurement error for long time series ($T = 800$) with randomly occurring outliers.

2.2.7 Standard error estimator performance

The figures in this section display the potential bias of the different method's AR parameter standard error (SE) estimators. Figure 64 pertains to data with no measurement error and short time series ($T = 80$), Figure 65 to data with measurement error and short time series, Figure 66 to data with no measurement error and medium-length time series ($T = 160$), Figure 67 to data with measurement error and medium-length time series, Figure 68 to data with no measurement error and long time series ($T = 800$), and Figure 69 to data with measurement error and long time series. We limit the presentation to randomly occurring outliers, and to models fitted without measurement error.

Each figure plots the average squared SE estimate of the AR parameter (y-axis) against the variance between AR parameter point estimates observed across simulation replications (x-axis). Outlier types are distinguished by panels, population-level AR effect sizes by columns and estimation methods are distinguished by rows. The turquoise lines delineate a one-to-one correspondence between both quantities, and deviations from it imply biased SE estimators in the sense that they do not recover the actual estimation uncertainty as estimated via the observed variance of the point estimates. Deviations in horizontal direction represent negative biases or underestimated uncertainties, which can inflate type 1 error rates, if SE estimates are used for hypothesis testing. Deviations in vertical direction represent positive biases of overestimated uncertainties, which can inflate type 2 error rates, if SE estimates are used for hypothesis testing.

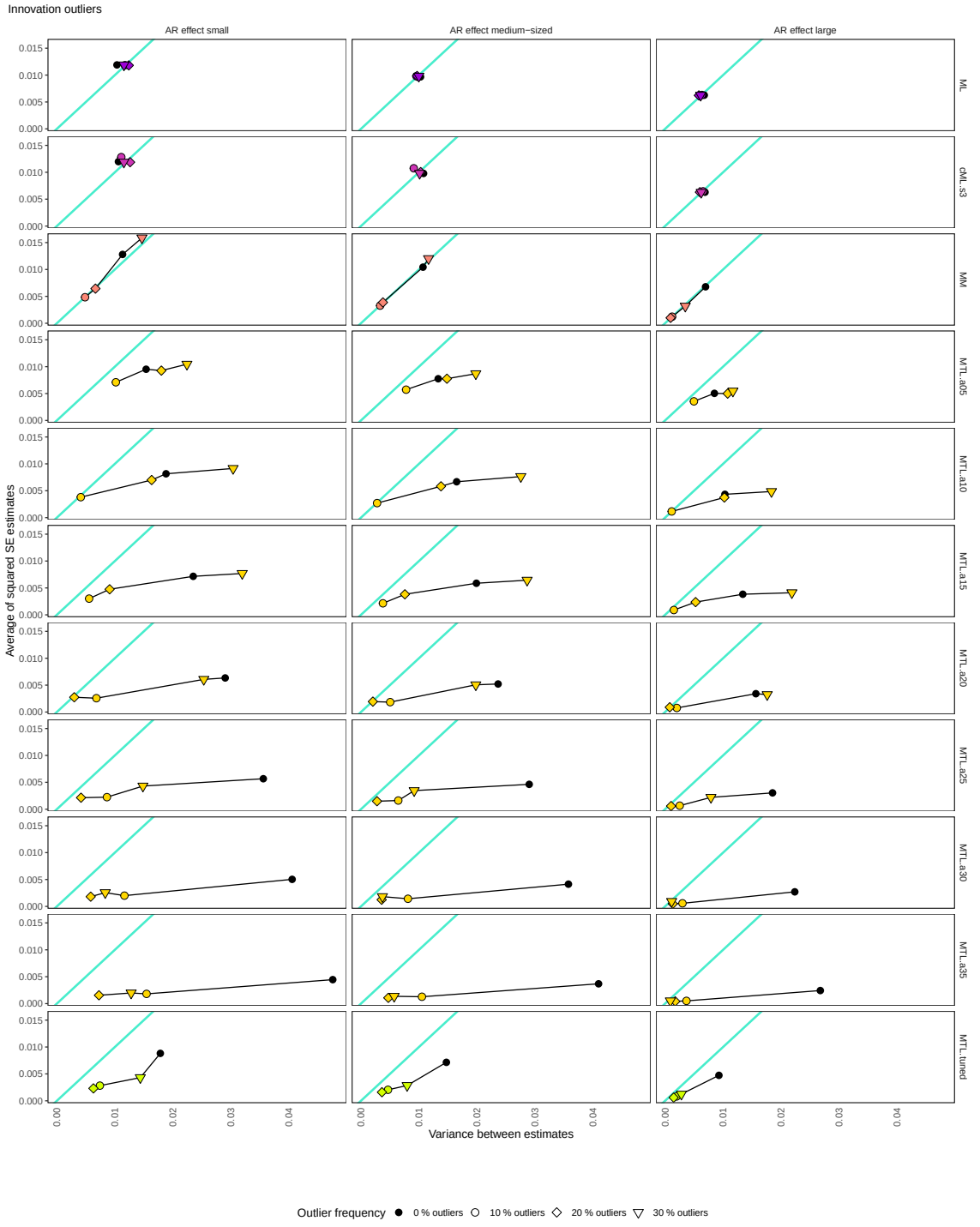


Figure 64

Standard error estimates for data with no measurement error for short time series ($T = 80$) with randomly occurring outliers and models fitted without measurement error.

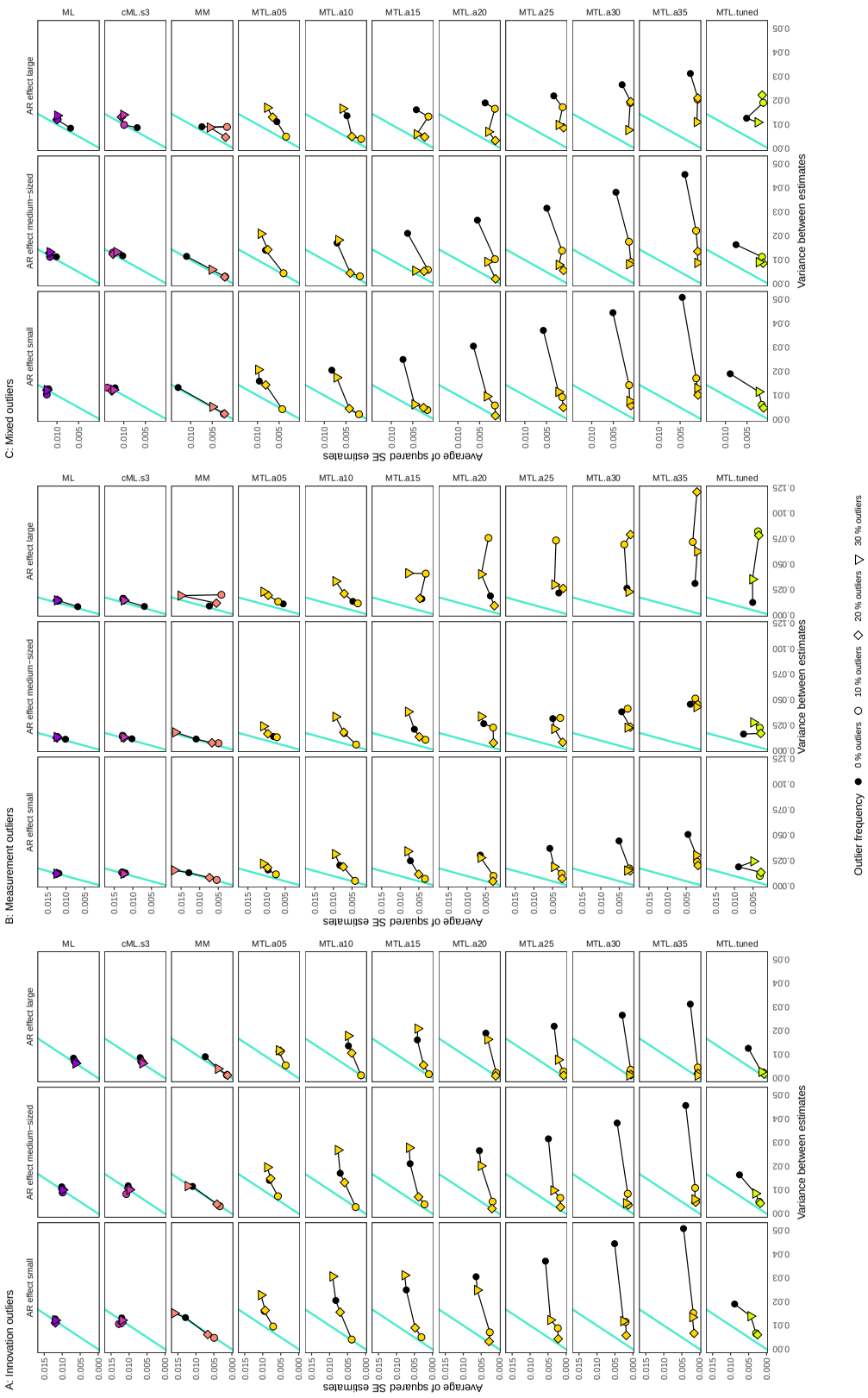


Figure 65

Standard error estimates for data with little measurement error for short time series ($T = 80$) with randomly occurring outliers and models fitted without measurement error.

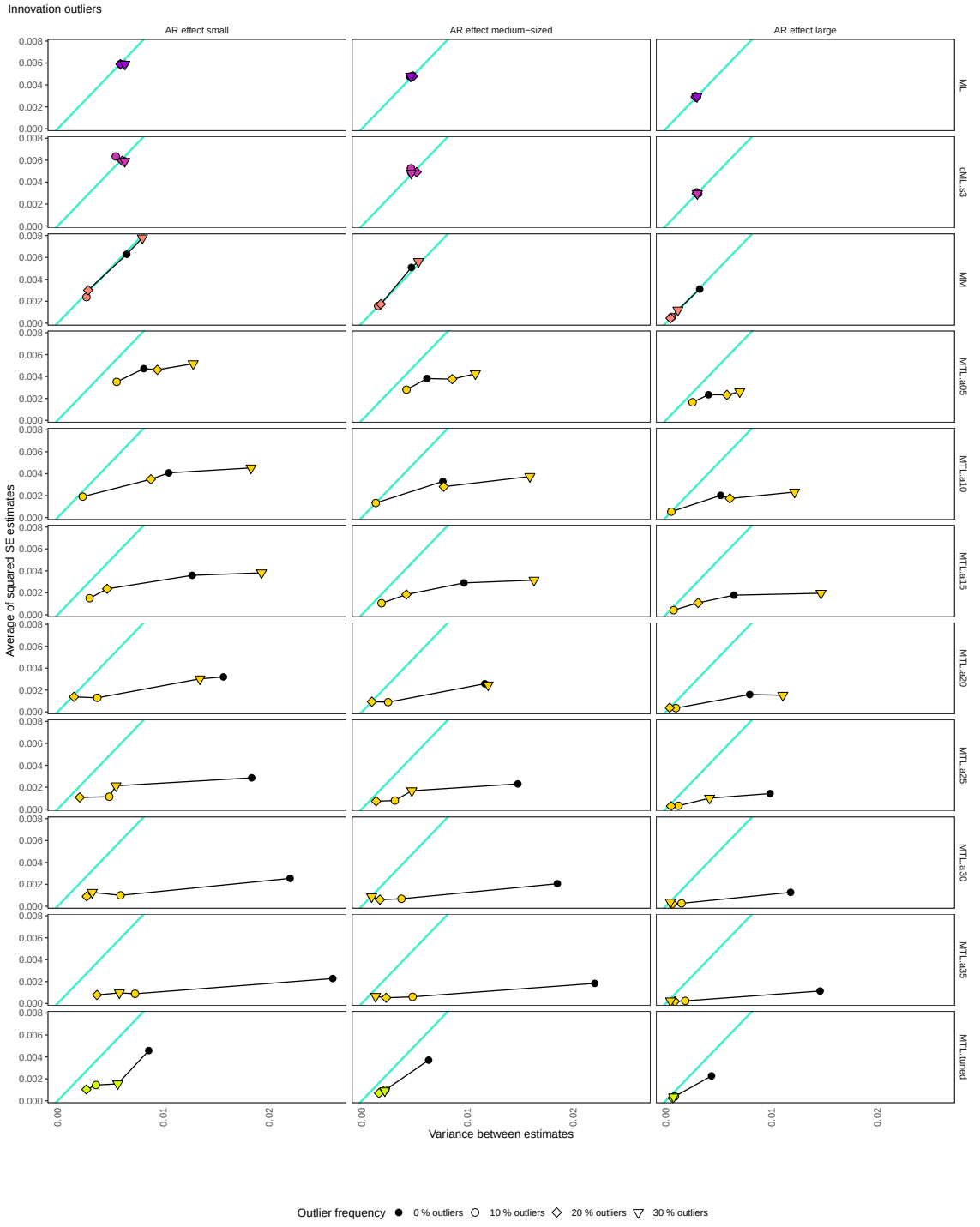


Figure 66

Standard error estimates for data with no measurement error for medium-length time series ($T = 160$) with randomly occurring outliers and models fitted without measurement error.

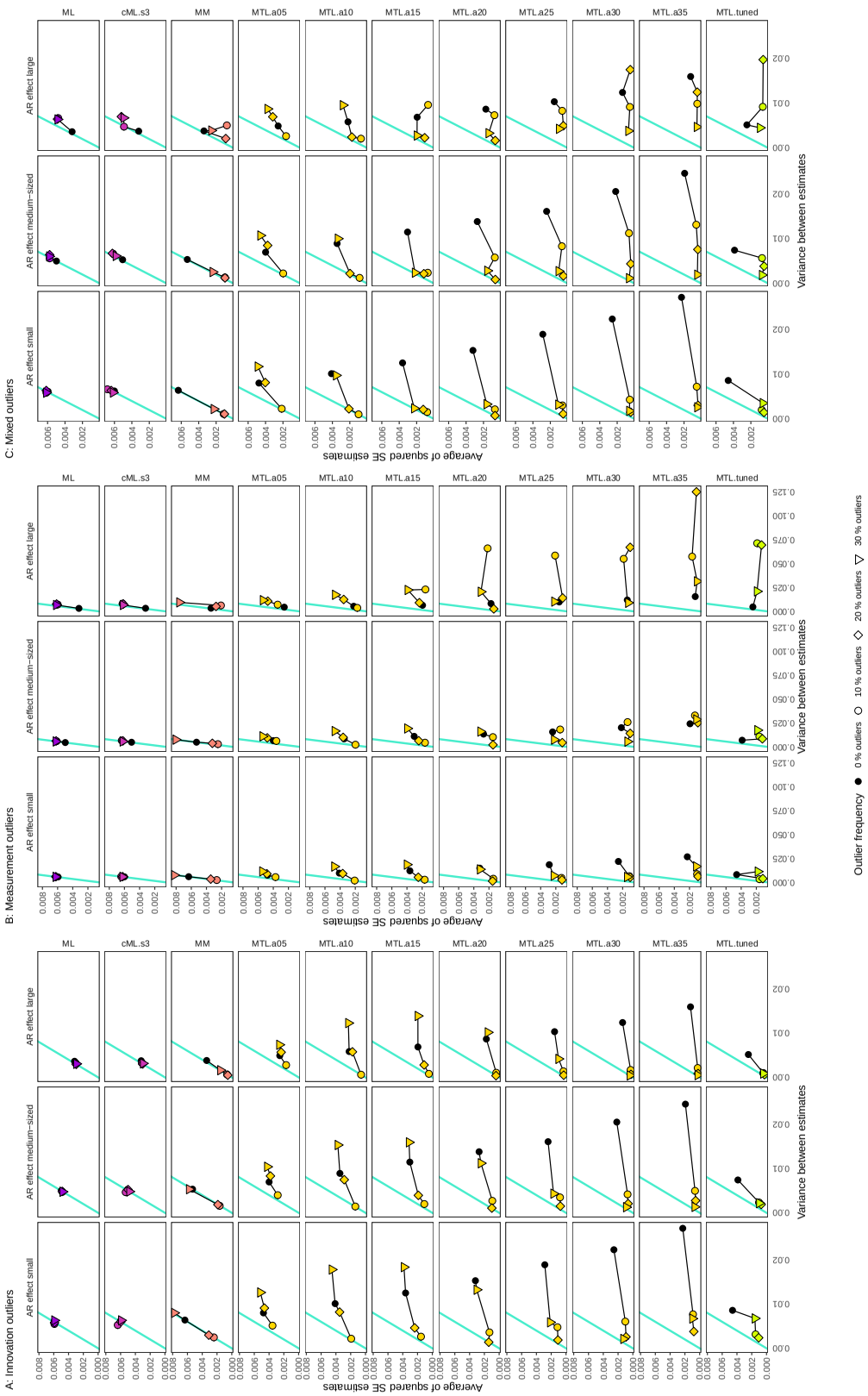


Figure 67

Standard error estimates for data with little measurement error for medium-length time series ($T = 160$) with randomly occurring outliers and models fitted without measurement error.

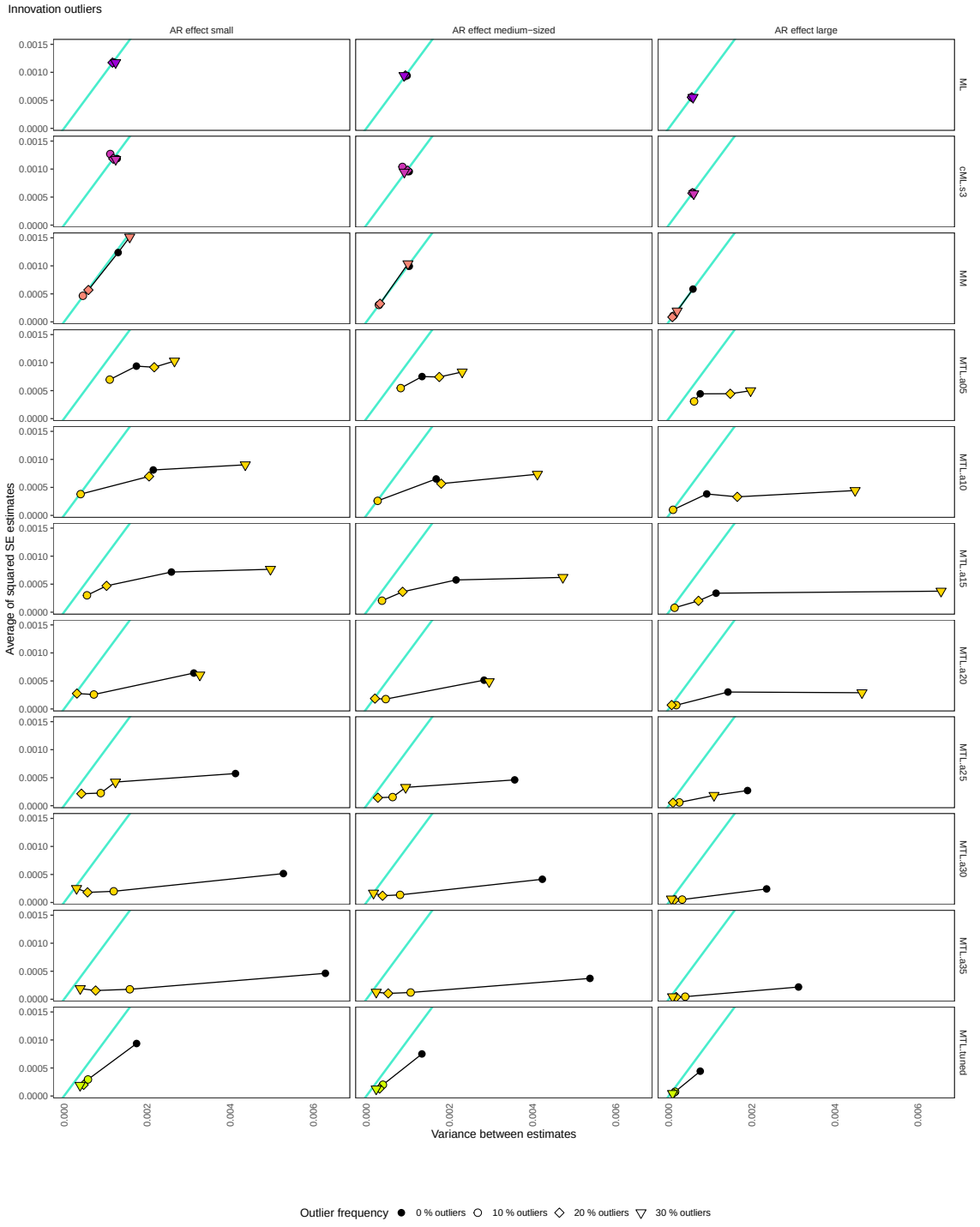


Figure 68

Standard error estimates for data with no measurement error for long time series ($T = 800$) with randomly occurring outliers and models fitted without measurement error.

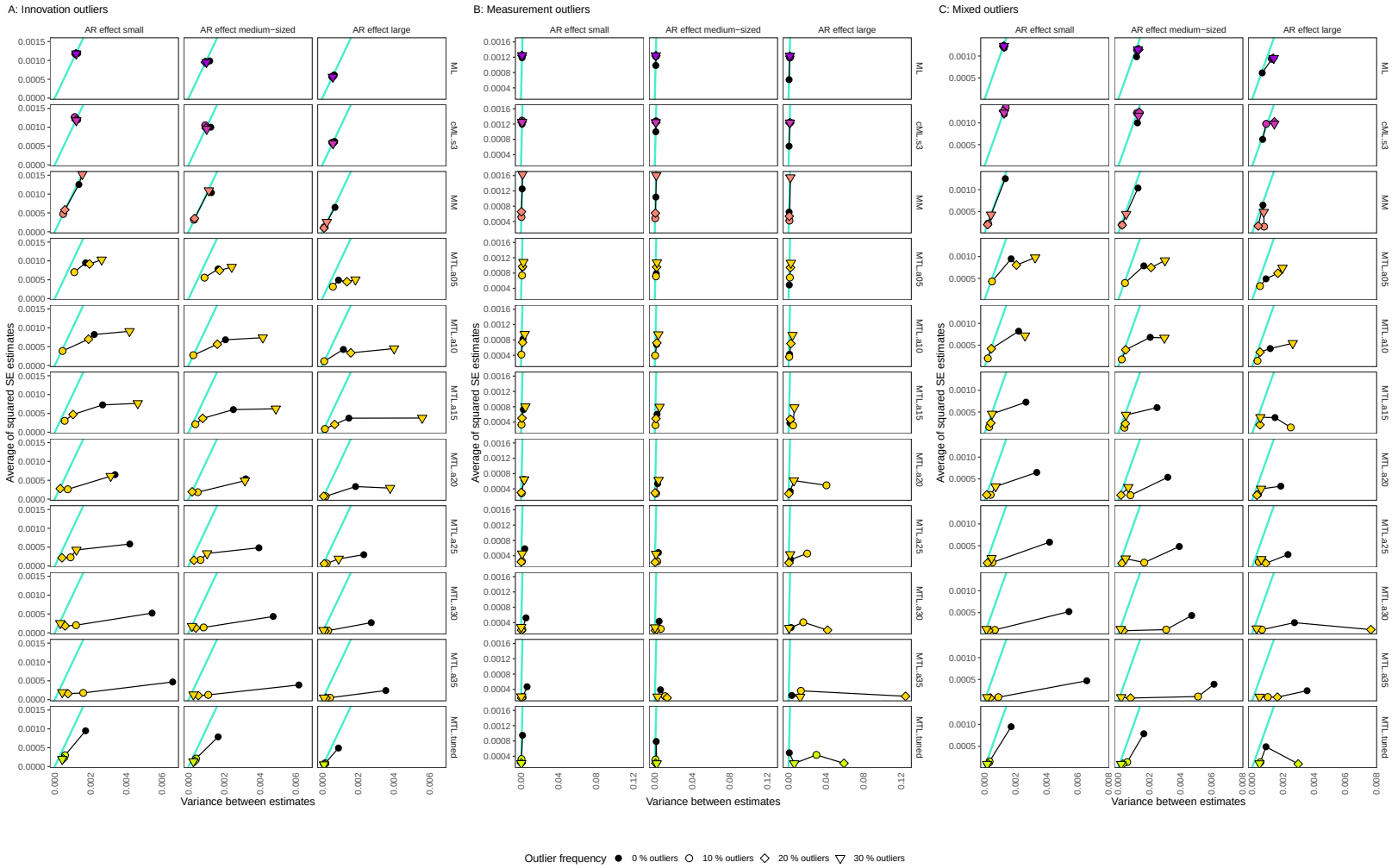


Figure 69

Standard error estimates for data with little measurement error for long time series ($T = 800$) with randomly occurring outliers and models fitted without measurement error.

3 Supplementary materials to the empirical illustration

3.1 Purpose

The purpose of this empirical illustration is to demonstrate the performance of the classical and robust methods considered in the manuscript under somewhat more realistic conditions in a real-world data set. To this end, we rely on the affect data from the gambling experiment (Ariens, 2023), which are described in detail in the manuscript and are available via an OSF repository (Ariens, 2023). We add innovation or measurement outliers to the observed data, and compare the performance of the various methods to the performance of classical AR(1) modeling of the original, uncontaminated data. All analysis code is available in a GitLab repository accessible via the Open Science Framework (OSF) repository *Supplementary materials for manuscript: Improved estimation of autoregressive models through contextual impulses and robust modeling* (Adolf & Ceulemans, 2023). The results are in line with the simulation study results.

3.2 Analysis

3.2.1 Outlier generation

We contaminate each time series with 10 % of outliers of each outlier type separately. This amount matches the smallest outlier frequency in the simulation study and is chosen to avoid too many (patchy) outliers, as this complicates their handling. As in the simulation study, outliers occur randomly, but their frequencies are implemented in an exact rather than a probabilistic manner. To enable this - and also to enable an exact implementation of the different trimming proportions of the MTL estimator - we use the first 140 observations per participant and thus discard the last 12 ones.[5] Also, the perturbing effect of each outlier indicator variable equals five times the observed overall SD of the individual affect time series in size. Unlike in the simulation study, however, there was a 50 % chance per measurement occasion that an outlier indicator would exert a negative instead of a positive effect.

To contaminate the data with measurement outliers, we first generate the binary outlier indicator variable, multiply it with the outlier effect of specific size and sign, and add the resulting outlier series to the observed data. Generating innovation outliers is more complicated and involves the following steps: We use the measurement outlier indicator variable values as impulses to a specific AR(1) system. This system has zero dynamic variance and a dynamic intercept moderated by the measurement outlier indicator variable. The AR effect of the system is set to the AR effect estimate obtained for the uncontaminated data per participant. The resulting time series resembles an impulse response function with multiple impulses at random time points, and is then used as an updated outlier indicator variable, although it is no longer binary. The updated outlier indicator variable is again multiplied with the outlier effect and the resulting series is added to the observed data. One can easily verify, using simulated data, that this procedure is equivalent to recursively generating data from an AR(1) system with innovation outliers. Note that by using the individual AR effect estimates, we make sure that the innovation outlier-based impulses feed forward according to each particpant's (estimated) individual dynamics.

3.2.2 Models fitted

To the contaminated data we fit all robust and classical AR(1) model variants employed in the simulation study, thus relying on $3-\sigma$ cML, MM MTL, and ML estimation. For MTL estimation we again use varying trimming proportions $\alpha = \{0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35\}$ and include tuning. The uncontaminated data are fitted with the classical AR(1) model only.

[5] Note that, starting from 152, 140 is the next smaller number for which the proportions of $\{0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35\}$ % correspond to whole numbers.

3.2.3 Assessing model performance

To assess model performance in this illustration, we calculate the observed root mean squared error (RMSE) of the AR estimates obtained under contamination with respect to the classical AR estimates obtained in the uncontaminated data. The observed RMSE is calculated across all individual AR estimates for each AR(1) model variant separately. It is then used to compare methods in the sense that smaller observed RMSEs mean better performance.

3.2.4 Implementation

We again rely on *R* (R Core Team, 2022), our own *simARO* package and its dependencies (see manuscript), and on the packages *parallel* (R Core Team, 2022) and *ggplot2* [with extensions; Wickham (2016)].

Figure 70 displays the participants' AR effect estimates obtained via robust and classical methods in the contaminated data (y-axis) as a function of classical estimates obtained in the uncontaminated data (x-axis). Contamination concerns innovation outliers in Panel A and measurement outliers in Panel B. Per panel, each tile displays the estimate pairs of all participants for one of the estimation methods considered under contamination. The methods are ordered such that their RMSE values increase from left to right and from top to bottom. How individual estimates deviate from the uncontaminated data estimates can be seen from the colored vertical line segments. The solid grey line thereby marks the identity line between contaminated uncontaminated data estimates.

Looking at Panel A, we see that both the MM estimator and the MTL estimator with the appropriate trimming proportion of 10 % perform best. The ML estimator performs below average (as can be read off the centered RMSE value added in brackets), and the MTL estimator with the two highest trimming proportion performs worst. Across all methods, contaminated data estimates deviate rather unsystematically from the benchmark results. Results thus seem to become *less variable* across participants with increasing method performance. This is in line with the results from the simulation study where we found that MSEs of the AR effect estimators were mainly variance-driven under innovation outliers.

Panel B shows the situation under measurement outliers instead. Here we see that it is again the MTL estimator with the appropriate trimming proportion that performs best (20 % due to vertical outliers *and* bad leverage points). There are a number of differences, though. First, the MTL estimator variants with larger trimming proportions perform above average, whereas the ones with the two smallest trimming proportions perform below average. The ML and $3 - \sigma$ cML estimators are clearly not performing well, and lie relatively far below average. Finally, we observe a rather systematic pattern of deviations across all methods, which is mainly determined by negative deviations of the contaminated data estimates from the identity line. Hence, under measurement outliers, individual results seem to become *less biased* and/or *less* individual results seem to become *biased* with increasing model performance. For the better performing methods we thus see roughly two clusters of solutions emerging: One cluster stretches across the whole range of benchmark estimate effect sizes along the x-axis, but is limited to smaller effect sizes for the contaminated data estimates along the y-axis. For the involved solutions bias is thus likely larger the larger the benchmark estimate. A second cluster concerns solutions with benchmark and comparison estimates of larger size, which lie relatively close to (or slightly above) the identity line. It seems to be the size of this cluster - and it's concentration around the identity line - that primarily determines method performance. That measurement outliers mainly cause bias problems is not a surprise and again in line with our simulation results.

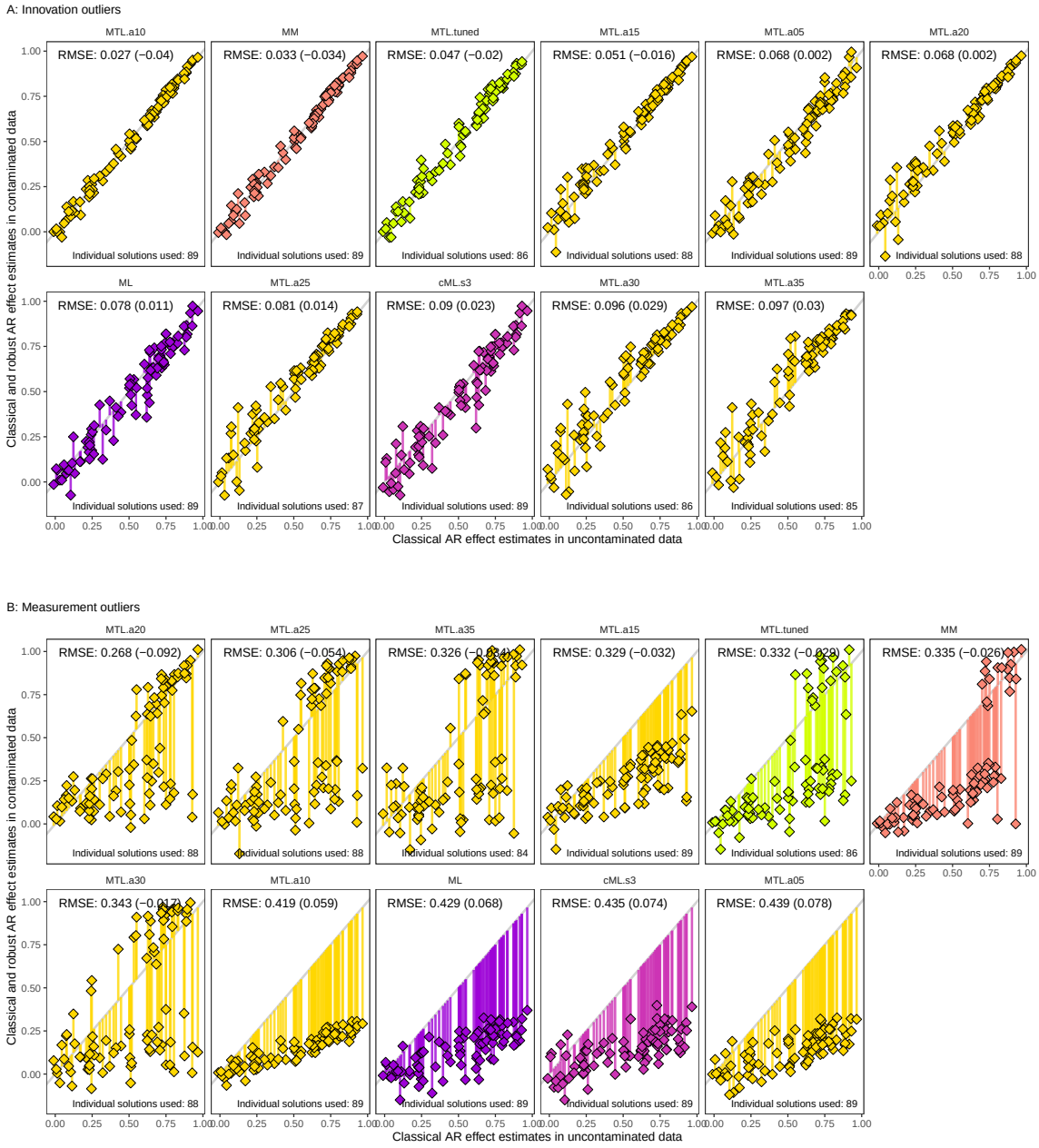


Figure 70

AR effect estimates from robust and classical methods in contaminated data (y-axis) as a function of classical estimates in uncontaminated data (x-axis). Results pertain to individual time series contaminated with innovation outliers (Panel A) or measurement outliers (Panel B). Estimation methods are distinguished by tiles and color, and are ranked according to their observed RMSE in ascending order. Observed RMSEs are displayed in the upper left corner of each tile, centered values are thereby added in brackets. A solid grey line marks the identity line between contaminated uncontaminated data estimates. Deviations of the contaminated data estimates from the identity line, and thus the uncontaminated data estimates are visualized via vertical line segments. Individual solutions are only included if the modeling approaches compared converged.

4 References

Adolf, J. K., & Ceulemans, E. (2023). *Supplementary materials for manuscript: Improved estimation of autoregressive models through contextual impulses and robust modeling.* Retrieved from https://osf.io/u5yda/

Ariens, S. (2023). *One does not simply correct for serial dependence: Supplementary materials.* Retrieved from https://osf.io/esmwa/

Aust, F., & Barth, M. (2022). *Papaja: Prepare reproducible APA journal articles with R Markdown.* Retrieved from https://github.com/crsh/papaja

Bodner, N., Tuerlinckx, F., Bosmans, G., & Ceulemans, E. (2021). Accounting for auto-dependency in binary dyadic time series data: A comparison of model- and permutation-based approaches for testing pairwise associations. *British Journal of Mathematical and Statistical Psychology.*

Chow, S.-M., Ho, M., Hamaker, E. L., & Dolan, C. V. (2010). Equivalence and Differences Between Structural Equation Modeling and State-Space Modeling Techniques. *Structural Equation Modeling: A Multidisciplinary Journal*, *17*(2), 303–332. https://doi.org/10.1080/10705511003661553

Crevits, R., & Croux, C. (2019). Robust estimation of linear state space models. *Communications in Statistics - Simulation and Computation*, *48*(6), 1694–1705. https://doi.org/10.1080/03610918.2017.1422752

Hardt, K., Hecht, M., Oud, J. H. L., & Voelkle, M. C. (2019). Where Have the Persons Gone? – An Illustration of Individual Score Methods in Autoregressive Panel Models. *Structural Equation Modeling: A Multidisciplinary Journal*, *26*(2), 310–323. https://doi.org/10.1080/10705511.2018.1517355

Harvey, A. C. (1989). *Forecasting, Structural Time Series Models and the Kalman Filter.* Cambridge, UK: Cambridge University Press.

Hunter, M. D. (2018). State Space Modeling in an Open Source, Modular, Structural Equation Modeling Environment. *Structural Equation Modeling: A Multidisciplinary Journal*, *25*(2), 307–324. https://doi.org/10.1080/10705511.2017.1369354

Maechler, M., Rousseeuw, P. J., Croux, C., Todorov, V., Ruckstuhl, A., Salibian-Barrera, M., … Palma, M. A. di. (2021). *Robustbase: Basic Robust Statistics R package.* Retrieved from https://cran.r-project.org/web/packages/robustbase/index.html

R Core Team. (2022). *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York. Retrieved from https://ggplot2.tidyverse.org

Yarkoni, T., & Westfall, J. (2017). Choosing Prediction Over Explanation in Psychology: Lessons From Machine Learning. *Perspectives on Psychological Science*, *12*(6), 1100–1122. https://doi.org/10.1177/1745691617693393