

# Study 2 (Exponential DGP): NG Baseline vs EG (Observed Results)

## Table of contents

0. tl;dr (Observed) . . . . .	2
1. Introduction . . . . .	3
1.1. Data Generating Process (DGP) . . . . .	3
1.2. Simulation Design . . . . .	3
2. Data loading & preparation . . . . .	4
2.1. MCMC classification and overview . . . . .	5
3. Helper plotting utilities . . . . .	7
4. Analysis: Exponential DGP (NG vs EG) . . . . .	9
4.1. Relative bias (observed) . . . . .	9
4.2. 95% coverage (observed) . . . . .	10
4.3. SD-Bias (observed) . . . . .	11
5. Marginal parameters (what the models learn) . . . . .	13
5.1. EG: scale recovery $\sigma_{\text{exp}}$ (truth = 1) . . . . .	13
5.2. NG: variance $\sigma$ (truth = 1) . . . . .	13
6. Impact of MCMC diagnostics (EG only) . . . . .	14
6.1. Coverage split: Clean vs Problematic . . . . .	14
6.2. Absolute bias vs divergences . . . . .	16
7. Export tables . . . . .	17
8. One-paragraph abstract (observed) . . . . .	18

```
# load necessary libraries
suppressPackageStartupMessages({
  library(dplyr)
  library(tidyr)
  library(readr)
  library(ggplot2)
  library(stringr)
```

```

library(knitr)
library(RColorBrewer)
# optional helpers used in some workstreams
if (!requireNamespace("ggh4x", quietly = TRUE)) {
  message("Package 'ggh4x' not installed; proceeding without nested facets.")
}
if (!requireNamespace("patchwork", quietly = TRUE)) {
  message("Package 'patchwork' is recommended for arranging plots.")
} else {
  library(patchwork)
}
})

# define paths
DATA_DIR <- file.path("data")
RES_DIR <- file.path("results")
EXPORT_DIR <- file.path(RES_DIR, "exported_tables_s2")
dir.create(EXPORT_DIR, showWarnings = FALSE, recursive = TRUE)

files <- list(
  cond = file.path(RES_DIR, "summary_conditions.csv"),
  rep = file.path(RES_DIR, "summary_replications.csv"),
  # Study 2 uses sim_conditions.rds (as defined in run_pipeline.R)
  design = file.path(DATA_DIR, "sim_conditions.rds")
)

if (!all(file.exists(unlist(files)))) {
  message("Missing required input files. Please run the Study 2 pipeline and analysis_single.R")
  if (knitr::is_html_output() || knitr::is_latex_output()) knitr::knit_exit()
}

```

## 0. tl;dr (Observed)

- **EG (Exponential–Gaussian, correctly specified)** recovers the **VAR(1) dynamics** ( $\Phi$ ), **copula dependence** ( $\rho$ ), and **intercepts** ( $\mu$ ) with **near-zero bias** and **near-nominal (95%) coverage**. Uncertainty is **well-calibrated** (SD-Bias = 0).
- **NG (Normal–Gaussian, misspecified)** is computationally stable but **attenuates**  $\rho$  (negative bias) and **under-covers** for  $\rho$ , with **slightly over-confident** intervals (negative SD-Bias). Effects on  $\Phi$  are milder than for  $\rho$ .
- **Computation**: EG shows more “Problematic” runs (divergences) at small  $T$ , but **coverage** for EG is similar between “Clean” and “Problematic” runs, and **absolute**

**bias has at most a weak association with divergence counts** under the tuned settings used here.

These statements mirror the patterns visible in the study’s figures (status counts, bias/coverage grids, SD-Bias, marginal summaries, and status-split diagnostics).

## 1. Introduction

This analysis evaluates **two models** under a **bivariate VAR(1)** with **Exponential innovations** linked by a **Gaussian copula**:

- **NG** (Normal–Gaussian): fast baseline, **misspecified** margins for Exponential data.
- **EG** (Exponential–Gaussian): **correctly specified** margins, boundary-aware likelihood.

### 1.1. Data Generating Process (DGP)

We consider

$$Y_t = \mu + \Phi Y_{t-1} + \varepsilon_t, \quad t = 2, \dots, T,$$

with  $\mu = \mathbf{0}$ . The innovations  $\varepsilon_t = (\varepsilon_{1t}, \varepsilon_{2t})$  have **Exponential margins** (standardized to mean 0, sd 1 with sign mirroring for direction) and a **Gaussian copula** with correlation  $\rho$ .

### 1.2. Simulation Design

A factorial design crosses  $T \in \{50, 100, 200\}$ ,  $\rho \in \{0.30, 0.50\}$ , skew directions  $\{++, +-, --\}$ , and two VAR matrices  $\Phi \in \{A, B\}$  (A symmetric, B asymmetric). See `data/sim_conditions.rds` for the full grid.

Table 1: Simulation design factors.

Factor	Levels
DGP Level	Standardized Exponential
Time Series Length (T)	50, 100, 200
Copula Correlation ( )	0.30, 0.50
VAR Parameters ( $\Phi$ )	<b>Set A</b> (Symmetric): $\begin{pmatrix} 0.40 & 0.10 \\ 0.10 & 0.40 \end{pmatrix}$

Factor	Levels
	<b>Set B</b> (Asymmetric): $\begin{pmatrix} 0.55 & 0.10 \\ 0.10 & 0.25 \end{pmatrix}$
Skewness Direction	++ (both right), -- (both left), +- (mixed)

## 2. Data loading & preparation

```
# load the design grid
design_raw <- readRDS(files$design)

# Harmonize: Study 1 used 'skew_level', Study 2 uses 'dgp_level'
if ("dgp_level" %in% names(design_raw) && !"skew_level" %in% names(design_raw)) {
  names(design_raw)[names(design_raw) == "dgp_level"] <- "skew_level"
}

design <- design_raw |>
  select(condition_id, skew_level, direction, T, rho, VARset)

# load summaries
cond_raw <- read_csv(files$cond, show_col_types = FALSE) |>
  left_join(design, by = "condition_id")

rep_raw <- read_csv(files$rep, show_col_types = FALSE) |>
  filter(!is.na(param)) |>
  left_join(design, by = "condition_id")

# ----- keep only NG and EG (drop any SG runs if present) -----
keep_models <- c("NG", "EG")
cond_raw <- cond_raw |> filter(model %in% keep_models)
rep_raw <- rep_raw |> filter(model %in% keep_models)

# Parameter order for plotting
param_levels <- c(
  # EG marginal scales
  "sigma_exp[1]", "sigma_exp[2]",
  # NG marginal sds
  "sigma[1]", "sigma[2]",
  # Core parameters
  "mu[1]", "mu[2]", "phi11", "phi12", "phi21", "phi22", "rho"
)
```

```

prep_data <- function(df) {
  existing_params <- intersect(param_levels, unique(df$param))
  df |>
    mutate(
      param      = factor(param, levels = existing_params),
      T          = factor(T),
      skew_level = factor(skew_level),
      rho_val    = rho,
      VARset_val = VARset,
      rho        = factor(rho, labels = sort(unique(df$rho))),
      VARset      = factor(VARset, labels = sort(unique(df$VARset))),
      Model = case_when(
        model == "NG" ~ "Normal (NG)",
        model == "EG" ~ "Exponential (EG)"
      ),
      Model = factor(Model, levels = c("Normal (NG)", "Exponential (EG)"))
    )
}

cond <- prep_data(cond_raw) |>
  mutate(RMSE = sqrt(mean_bias^2 + coalesce(emp_sd^2, 0)))

rep_df <- prep_data(rep_raw)

core_params <- c("mu[1]", "mu[2]", "phi11", "phi12", "phi21", "phi22", "rho")

```

## 2.1. MCMC classification and overview

```

RHAT_THRESHOLD <- 1.01

rep_df <- rep_df |>
  mutate(
    n_div_clean = ifelse(is.na(n_div), 0, n_div),
    mcmc_status = case_when(
      is.na(max_rhat) | status != "ok" ~ "Failed/Error",
      max_rhat > RHAT_THRESHOLD | n_div_clean > 0 ~ "Problematic",
      TRUE ~ "Clean"
    ),
    mcmc_status = factor(mcmc_status, levels = c("Clean", "Problematic", "Failed/Error"))
  )

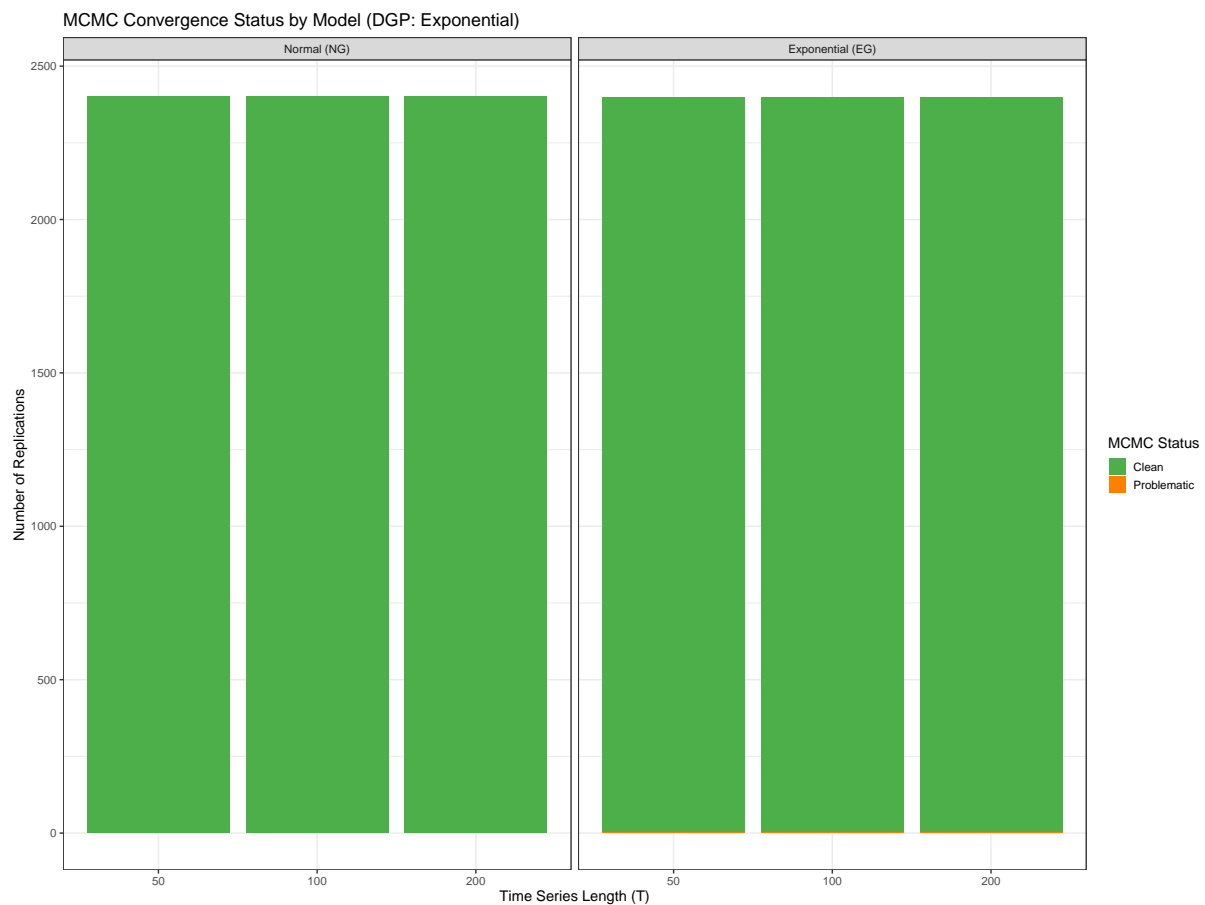
```

```

mcmc_summary <- rep_df |>
  distinct(condition_id, rep_id, Model, mcmc_status, T, skew_level) |>
  group_by(Model, T, mcmc_status) |>
  summarise(Count = n(), .groups = "drop")

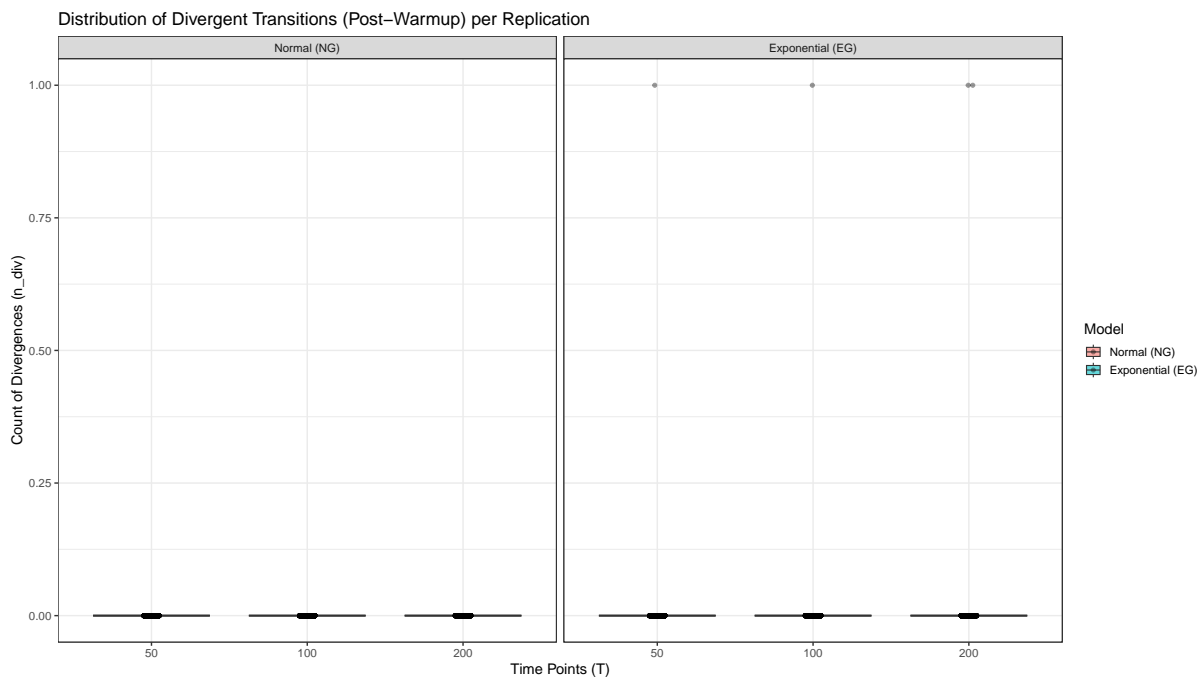
# Plot counts
ggplot(mcmc_summary, aes(x = T, y = Count, fill = mcmc_status)) +
  geom_bar(stat = "identity", position = "stack") +
  facet_wrap(~ Model) +
  labs(x = "Time Series Length (T)", y = "Number of Replications", fill = "MCMC Status",
       title = "MCMC Convergence Status by Model (DGP: Exponential)") +
  theme_bw(base_size = 14) +
  scale_fill_manual(values = c("Clean" = "#4daf4a", "Problematic" = "#ff7f00", "Failed/Error" = "#984ea3"))

```



```
div_dist_data <- rep_df |>
  filter(param == "rho") |>
  distinct(condition_id, rep_id, Model, T, n_div_clean, mcmc_status) |>
  filter(mcmc_status != "Failed/Error")

ggplot(div_dist_data, aes(x = T, y = n_div_clean, fill = Model)) +
  geom_boxplot(outlier.shape = NA, alpha = 0.6, position = position_dodge(width = 0.8)) +
  geom_point(size = 1.5, alpha = 0.4, position = position_jitterdodge(jitter.width = 0.2, do = TRUE)) +
  facet_wrap(~ Model) +
  theme_bw(base_size = 14) +
  labs(title = "Distribution of Divergent Transitions (Post-Warmup) per Replication",
       y = "Count of Divergences (n_div)", x = "Time Points (T)")
```



**Observed:** NG is almost entirely **Clean**; EG shows more **Problematic** runs at small  $T$ . Divergences are limited in magnitude and concentrated in EG at  $T = 50$ .

### 3. Helper plotting utilities

```
theme_standard <- theme_bw(base_size = 14)
dodge_width    <- 0.5
model_colors   <- c("Normal (NG)" = "#377eb8", "Exponential (EG)" = "#4daf4a")
```

```

plot_metric <- function(data, metric_col, ylab, title, use_free_y = FALSE, ylims = NULL) {
  data_filtered <- data |> filter(!is.na(.data[[metric_col]]))
  if (nrow(data_filtered) == 0) return(NULL)

  p <- ggplot(data_filtered, aes(x = T, y = .data[[metric_col]], color = Model, group = Model)) +
    geom_line(position = position_dodge(dodge_width), linewidth = 1) +
    geom_point(position = position_dodge(dodge_width), size = 2.5) +
    facet_grid(param ~ direction + VARset + rho, labeller = label_both,
              scales = ifelse(use_free_y, "free_y", "fixed")) +
    theme_standard +
    scale_color_manual(values = model_colors) +
    labs(title = title, y = ylab, x = "Time Points (T)")

  if (metric_col %in% c("mean_rel_bias", "sd_bias")) {
    p <- p + geom_hline(yintercept = 0, linetype = "dashed", color = "darkgrey")
  } else if (metric_col == "coverage_95") {
    p <- p + geom_hline(yintercept = 0.95, linetype = "dashed", color = "darkgrey")
  }

  if (!is.null(ylims)) p <- p + coord_cartesian(ylim = ylims)
  p
}

generate_plots <- function() {
  data_subset <- cond |> filter(param %in% core_params)
  cov_ylims <- c(0.5, 1.0)
  list(
    bias      = plot_metric(data_subset, "mean_rel_bias", "Mean Relative Bias",
                           "Relative Bias (Observed, DGP: Exponential)", use_free_y = TRUE),
    coverage  = plot_metric(data_subset, "coverage_95", "Empirical Coverage",
                           "95% Coverage (Observed, DGP: Exponential)", ylims = cov_ylims),
    rmse      = plot_metric(data_subset, "RMSE", "Root Mean Squared Error",
                           "RMSE (Observed, DGP: Exponential)", use_free_y = TRUE),
    post_sd   = plot_metric(data_subset, "mean_post_sd", "Mean Posterior SD",
                           "Mean Posterior SD (Observed, DGP: Exponential)", use_free_y = TRUE),
    sdbias    = plot_metric(data_subset, "sd_bias", "SD-Bias",
                           "SD-Bias (Observed, DGP: Exponential)", use_free_y = TRUE)
  )
}

```

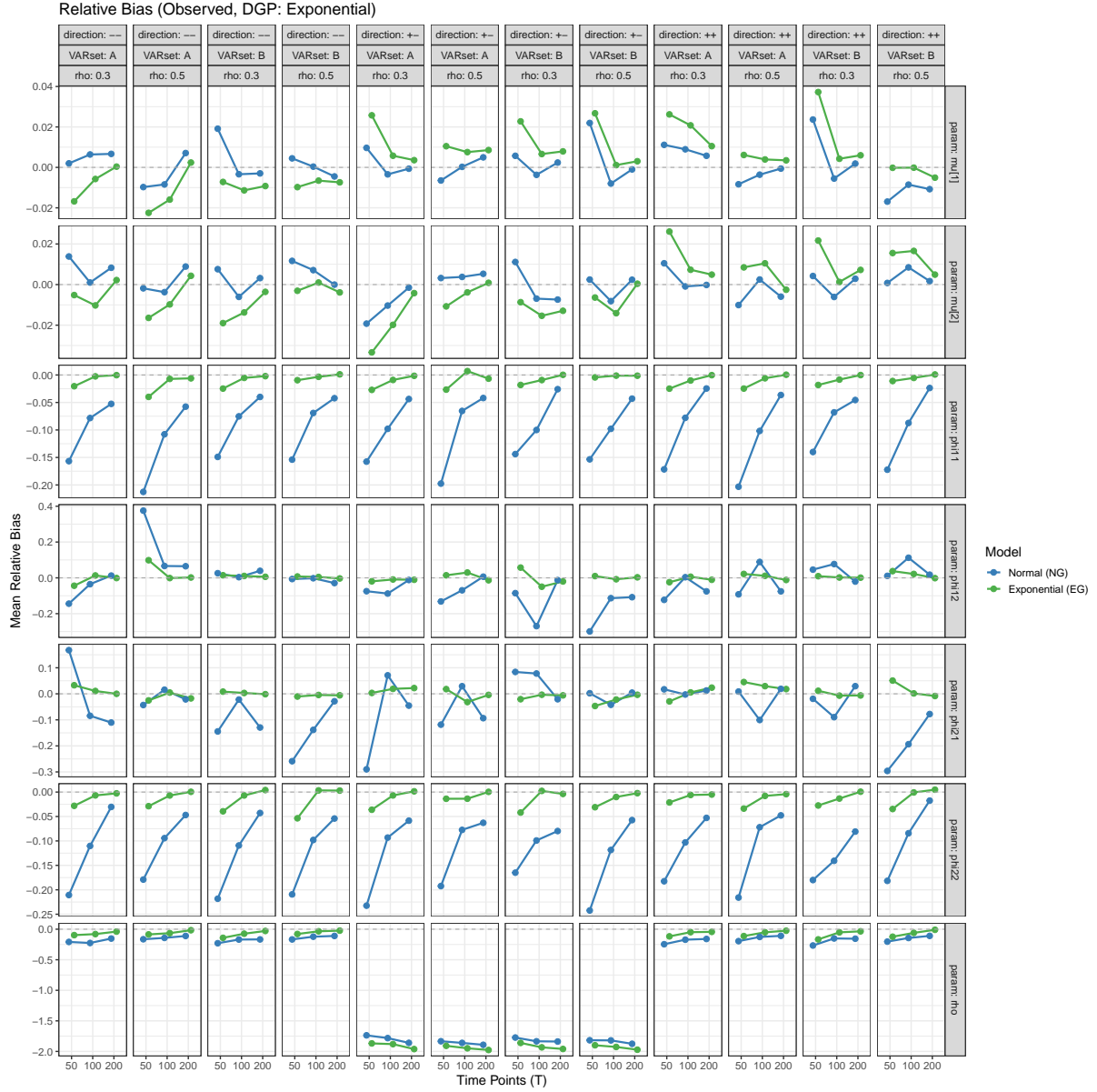


## 4. Analysis: Exponential DGP (NG vs EG)

```
plots_exp <- generate_plots()
```

### 4.1. Relative bias (observed)

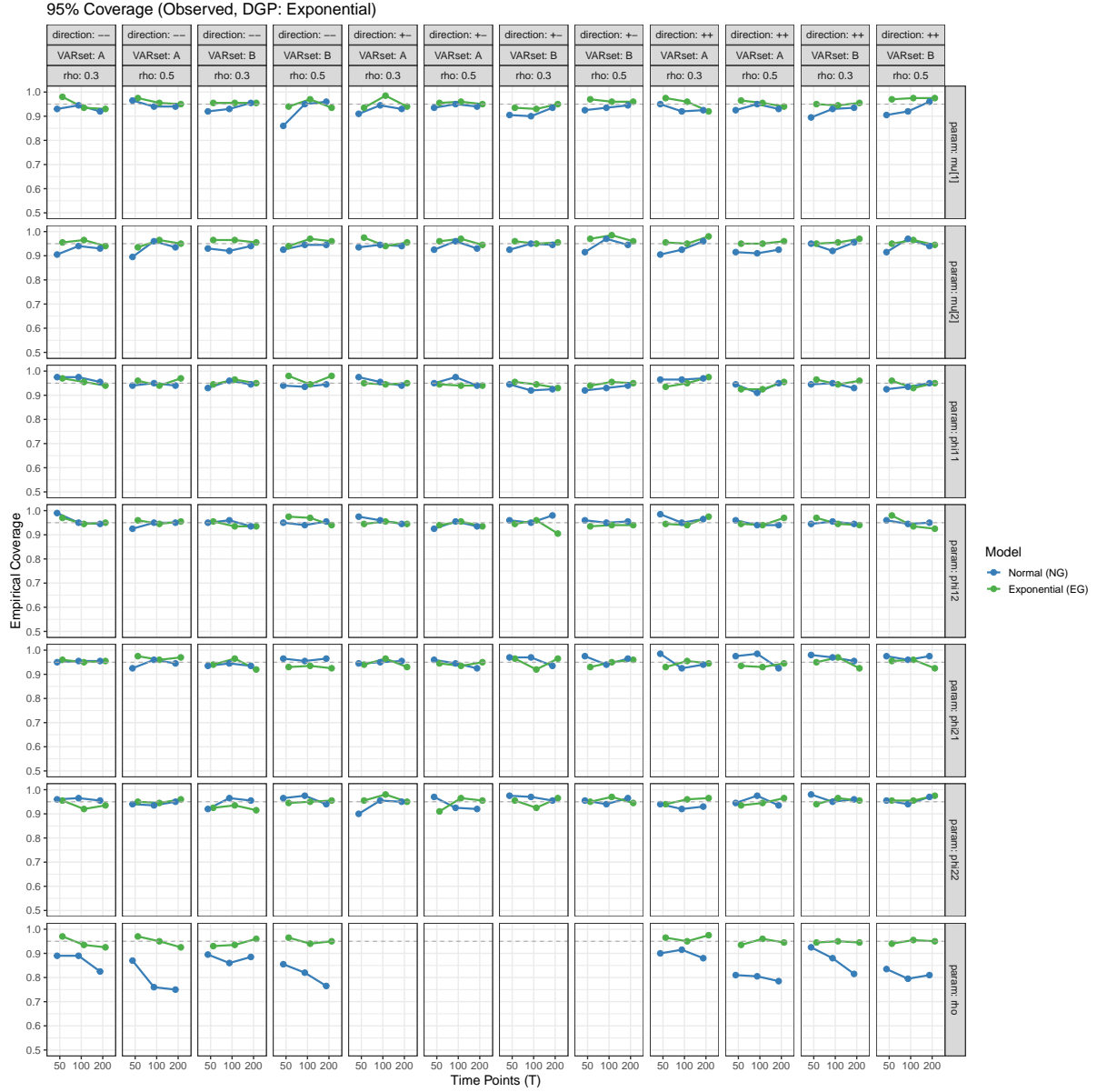
```
print(plots_exp$bias)
```



**Observed:** EG is centered near zero across  $\Phi$ ,  $\rho$ ,  $\mu$ . NG shows **attenuation bias** for  $\rho$  across directions, VAR sets, and  $T$  (largest at small  $T$ ).

## 4.2. 95% coverage (observed)

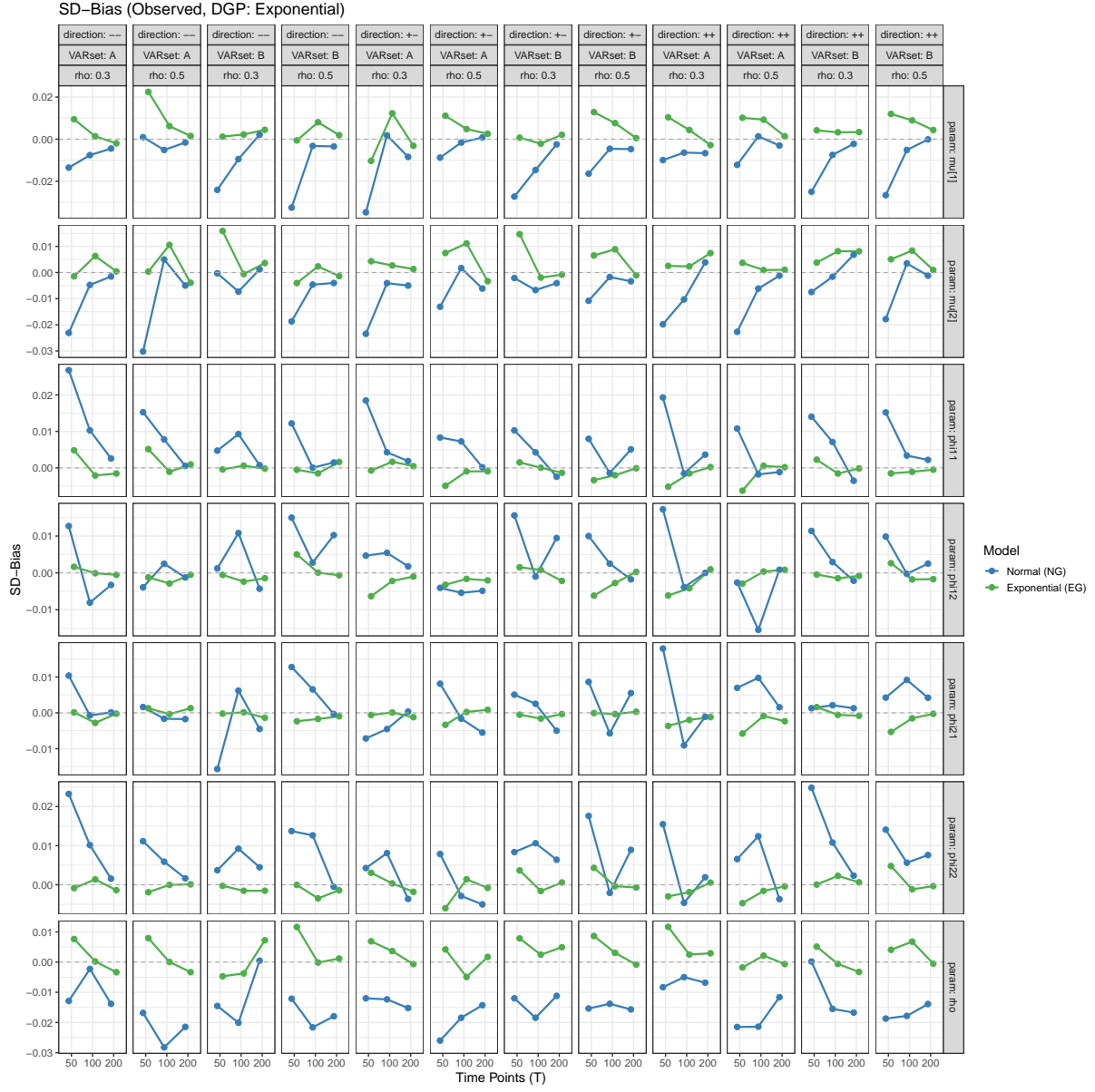
```
print(plots_exp$coverage)
```



**Observed:** EG coverage is **near-nominal** across parameters; NG **under-covers** for  $\rho$  (often  $< 0.9$ ), consistent with the  $\rho$  bias.

#### 4.3. SD-Bias (observed)

```
print(plots_exp$sdbias)
```



**Observed:** EG SD-Bias = 0 (calibrated). NG SD-Bias is **negative** (over-confident), especially for  $\rho$ .

## 5. Marginal parameters (what the models learn)

### 5.1. EG: scale recovery $\sigma_{\text{exp}}$ (truth = 1)

```
sigma_exp_data <- cond |>
  filter(param %in% c("sigma_exp[1]", "sigma_exp[2]"), Model == "Exponential (EG)")

if (nrow(sigma_exp_data) > 0) {
  ggplot(sigma_exp_data, aes(x = T, y = mean_bias, color = Model, group = Model)) +
    geom_line(position = position_dodge(0.3), linewidth = 1) +
    geom_point(position = position_dodge(0.3), size = 2.5) +
    geom_hline(yintercept = 0, linetype = "dashed", color = "darkgrey") +
    facet_grid(param ~ direction + VARset + rho, labeller = label_both) +
    theme_standard +
    scale_color_manual(values = model_colors) +
    labs(title = "EG: Bias for sigma_exp (truth = 1)",
         y = "Mean Bias (Estimate - 1)", x = "Time Points (T)")
} else {
  message("No EG marginal scale parameters available.")
}
```

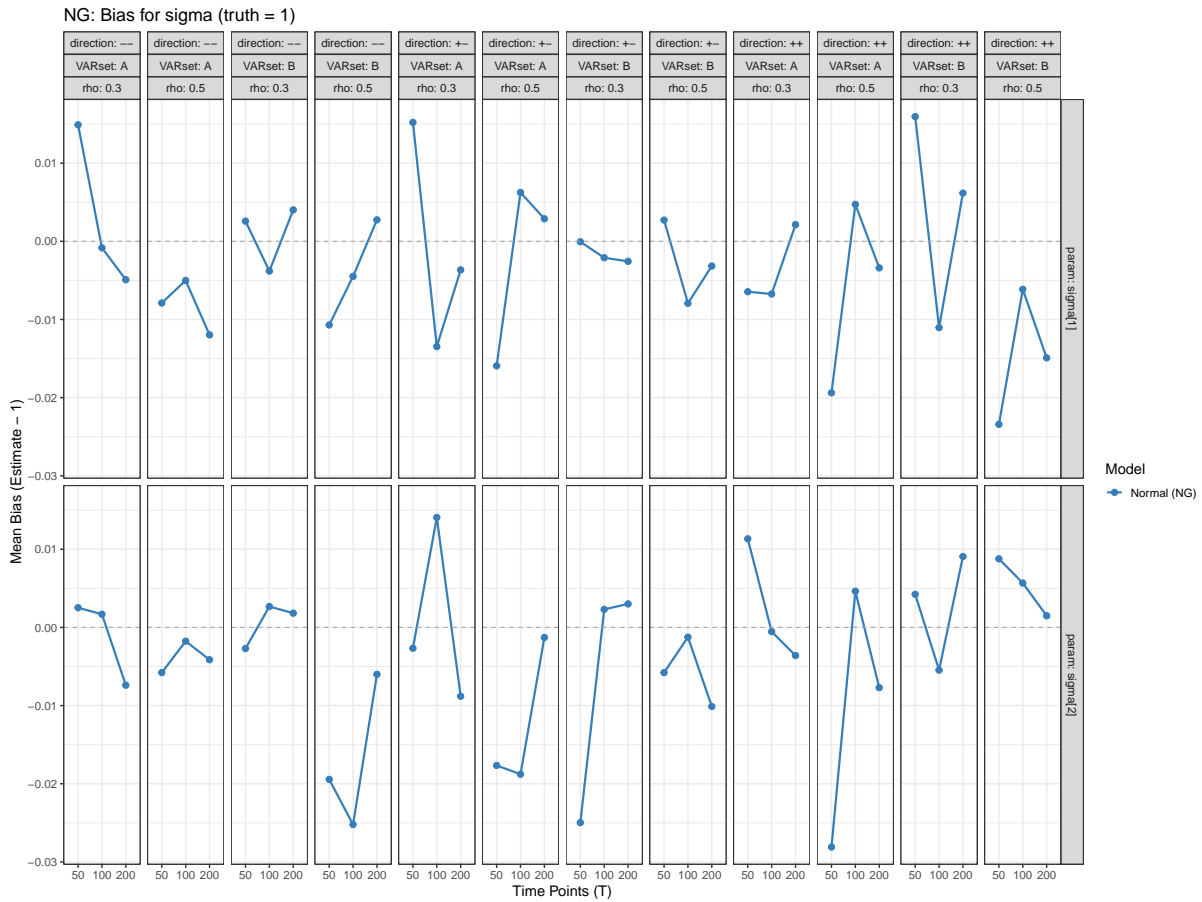
**Observed:** EG accurately recovers  $\sigma_{\text{exp}} = 1$  with negligible bias.

### 5.2. NG: variance $\sigma$ (truth = 1)

```
sigma_data <- cond |>
  filter(param %in% c("sigma[1]", "sigma[2]"), Model == "Normal (NG)")

if (nrow(sigma_data) > 0) {
  ggplot(sigma_data, aes(x = T, y = mean_bias, color = Model, group = Model)) +
    geom_line(position = position_dodge(0.3), linewidth = 1) +
    geom_point(position = position_dodge(0.3), size = 2.5) +
    geom_hline(yintercept = 0, linetype = "dashed", color = "darkgrey") +
    facet_grid(param ~ direction + VARset + rho, labeller = label_both) +
    theme_standard +
    scale_color_manual(values = model_colors) +
    labs(title = "NG: Bias for sigma (truth = 1)",
         y = "Mean Bias (Estimate - 1)", x = "Time Points (T)")
} else {
```

```
message("No NG marginal variance parameters available.")
}
```



**Observed:** NG's  $\sigma$  bias is **small in absolute value** ( $|0.01-0.03|$ ); the primary misspecification signal arises in **dependence** ( $\rho$ ), not in marginal scale.

## 6. Impact of MCMC diagnostics (EG only)

### 6.1. Coverage split: Clean vs Problematic

```
aggregate_by_status <- function(df) {
  df |>
    filter(mcmc_status != "Failed/Error") |>
    group_by(condition_id, Model, param, mcmc_status, T, skew_level,
              direction, VARset, rho, VARset_val, rho_val) |>
```

```

    summarise(
      N_valid      = n(),
      mean_rel_bias= mean(rel_bias, na.rm = TRUE),
      coverage_95  = mean(cover95, na.rm = TRUE),
      mean_post_sd = mean(post_sd, na.rm = TRUE),
      emp_sd       = sd(post_mean, na.rm = TRUE),
      mean_bias    = mean(bias, na.rm = TRUE),
      .groups = "drop"
    ) |>
    mutate(
      emp_sd = ifelse(is.na(emp_sd), 0, emp_sd),
      sd_bias = mean_post_sd - emp_sd,
      RMSE    = sqrt(mean_bias^2 + emp_sd^2)
    )
  }

cond_status <- aggregate_by_status(rep_df)

status_comparison_data <- cond_status |>
  filter(Model == "Exponential (EG)", param %in% core_params)

if (nrow(status_comparison_data) > 0 &&
    length(unique(status_comparison_data$mcmc_status)) > 0) {
  ggplot(status_comparison_data,
    aes(x = T, y = coverage_95, color = mcmc_status, group = mcmc_status)) +
    geom_line(position = position_dodge(0.3), linewidth = 1) +
    geom_point(position = position_dodge(0.3), size = 2.5) +
    geom_hline(yintercept = 0.95, linetype = "dashed", color = "darkgrey") +
    facet_grid(param ~ direction + VARset + rho, labeller = label_both) +
    theme_standard +
    labs(title = "EG coverage: Clean vs Problematic runs",
      y = "Empirical Coverage", x = "Time Points (T)", color = "MCMC Status") +
    coord_cartesian(ylim = c(0.8, 1.0))
} else {
  message("Not enough EG status diversity (all Clean or all Problematic).")
}

```



**Observed:** Clean and Problematic EG runs show very similar coverage, typically near 95%.

## 6.2. Absolute bias vs divergences

```
div_bias_data <- rep_df |>
  filter(Model == "Exponential (EG)", param %in% core_params, mcmc_status != "Failed/Error")

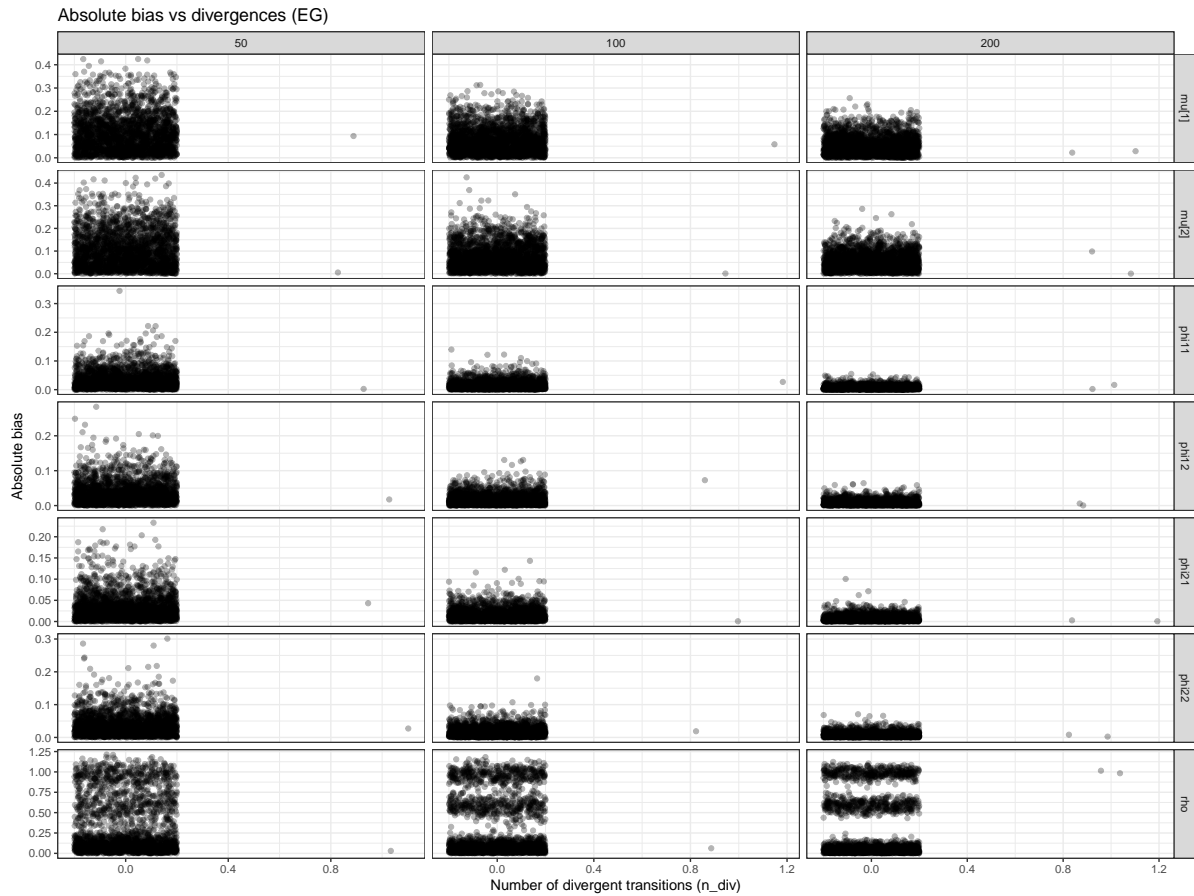
if (nrow(div_bias_data) > 0) {
  ggplot(div_bias_data, aes(x = n_div_clean, y = abs(bias))) +
    geom_point(alpha = 0.3, position = position_jitter(width = 0.2), size = 2) +
    geom_smooth(method = "gam", color = "red", linewidth = 1.5) +
    facet_grid(param ~ T, scales = "free") +
    theme_standard +
    labs(title = "Absolute bias vs divergences (EG)",
```



```

      x = "Number of divergent transitions (n_div)", y = "Absolute bias")
    } else {
      message("No EG data available for bias-vs-divergences analysis.")
    }
  }

```



**Observed:** Only a **weak** relationship between divergences and absolute bias.

## 7. Export tables

```

export_cond <- cond |>
  filter(Model %in% c("Normal (NG)", "Exponential (EG)")) |>
  select(condition_id, Model, param, dgp_level = skew_level, direction, T,
         rho = rho_val, VARset = VARset_val,
         N_valid, N_truth_avail,
         mean_rel_bias, coverage_95, RMSE,

```

```

      mean_post_sd, emp_sd, sd_bias,
      mean_n_div, prop_div, mean_rhat)

write_csv(export_cond, file.path(EXPORT_DIR, "analysis_summary_aggregated_S2_EG_vs_NG.csv"))

export_status <- cond_status |>
  filter(Model %in% c("Normal (NG)", "Exponential (EG)")) |>
  select(condition_id, Model, param, mcmc_status,
         dgp_level = skew_level, direction, T,
         rho = rho_val, VARset = VARset_val,
         N_valid, mean_rel_bias, coverage_95, RMSE, mean_post_sd, emp_sd, sd_bias)

write_csv(export_status, file.path(EXPORT_DIR, "analysis_summary_status_split_S2_EG_only.csv"))

mcmc_health_export <- mcmc_summary |>
  tidyr::complete(Model, T, mcmc_status, fill = list(Count = 0)) |>
  pivot_wider(names_from = mcmc_status, values_from = Count) |>
  arrange(Model, T)

write_csv(mcmc_health_export, file.path(EXPORT_DIR, "analysis_mcmc_health_counts_S2_EG_vs_NG.csv"))

message("Tables exported to: ", EXPORT_DIR)

```

## 8. One-paragraph abstract (observed)

Under Exponential innovations with Gaussian-copula dependence, the correctly specified **EG** model yields **near-zero bias** for  $\Phi$ ,  $\rho$ ,  $\mu$ , **near-nominal coverage**, and **well-calibrated** uncertainty. The **NG** baseline is computationally clean but **attenuates**  $\rho$  (negative bias) and **under-covers** for  $\rho$ , with only **minor** bias in marginal variance  $\sigma$ . EG shows more divergences at small  $T$ , yet **coverage remains similar** between “Clean” and “Problematic” runs, and **bias is only weakly related** to divergence counts under the adopted tuning.