

Study 2: Exponential Innovations — NG vs EG

Table of contents

| | |
|---|-----------|
| 1 0 Summary | 2 |
| 2 1 Background and simulation design | 3 |
| 2.1 1.1 Data-generating process | 3 |
| 2.2 1.2 Simulation factors | 3 |
| 3 2 Data loading and preprocessing | 3 |
| 4 3 MCMC diagnostics overview | 5 |
| 5 4 Core parameter recovery | 8 |
| 5.1 4.1 Condition-level aggregation | 8 |
| 5.2 4.2 Relative bias | 9 |
| 5.3 4.3 95% coverage | 10 |
| 5.4 4.4 SD-bias | 11 |
| 6 5 Marginal scale parameters | 12 |
| 6.1 5.1 EG: σ_{exp} (truth = 1) | 12 |
| 6.2 5.2 NG: σ (truth = 1) | 13 |
| 7 6 EG only: role of MCMC pathologies | 14 |
| 7.1 6.1 Coverage split by MCMC status | 14 |
| 7.2 6.2 Absolute bias vs divergences | 15 |
| 8 7 Export tables | 16 |
| 9 8 Abstract | 16 |

1 0 Summary

This report evaluates a bivariate VAR(1) with standardized Exponential innovations under two copula-VAR specifications:

- **NG (Normal–Gaussian):** Gaussian copula with *Normal* marginal innovation model (misspecified under Exponential DGP).
- **EG (Exponential–Gaussian):** Gaussian copula with *Exponential* marginal innovation model (correct family under the Study 2 DGP), implemented via a boundary-aware reparameterization.

Main empirical patterns (conditional on successful fits):

1. **Dynamics and intercepts.** Both models recover μ and Φ with small bias at moderate-to-large T . Differences between NG and EG are modest for Φ , relative to the dependence parameter.
2. **Copula dependence.** For same-direction skew settings (++ and --), EG is approximately unbiased for ρ with close-to-nominal coverage, whereas NG exhibits attenuation in $|\rho|$ and under-coverage for ρ .
3. **Computation.** EG shows more divergent transitions at small T . Within EG, empirical coverage is typically similar between “Clean” and “Problematic” (divergent or high- \hat{R}) fits, and absolute bias is weakly associated with divergence counts.

! Copula sign under *mirroring* (critical for interpreting ρ in mixed-direction cells)

In Study 2, “left-skew” margins are produced by *mirroring* standardized Exponential innovations (multiplying by -1). This is a **monotone decreasing** transformation of the latent Uniform variates used by the Gaussian copula generator. For a Gaussian copula, flipping exactly one margin maps $\rho \mapsto -\rho$.

Consequently, in mixed-direction settings (e.g., +–), the **effective copula parameter on the PIT scale** is

$$\rho_{\text{eff}} = s_1 s_2 \rho, \quad s_j \in \{+1, -1\},$$

where $s_j = -1$ indicates a mirrored (left-skew) margin and $s_j = +1$ indicates an unmirrored (right-skew) margin.

All model-based diagnostics for ρ in this report therefore use ρ_{eff} as the ground truth. Without this adjustment, mixed-direction panels will display an apparent “sign reversal” (relative bias near -2) and near-zero coverage for ρ , which is a *parameterization mismatch*, not a failure of the sampler.

i Relative bias for μ

Because $\mu = 0$ in the DGP, “relative bias” is reported as **absolute bias** for μ (i.e., $\hat{\mu} - 0$), using the convention `rel_bias = bias` when $|\text{truth}|$ is near 0.

2 1 Background and simulation design

2.1 1.1 Data-generating process

We consider

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

with $\mu = \mathbf{0}$. Innovations $\varepsilon_t = (\varepsilon_{1t}, \varepsilon_{2t})$ have standardized Exponential margins (mean 0, sd 1), with optional sign mirroring to induce left skew, and a Gaussian copula dependence structure.

2.2 1.2 Simulation factors

The design crosses:

- $T \in \{50, 100, 200\}$
- $\rho \in \{0.30, 0.50\}$ (input copula correlation before mirroring; $\rho_{\text{eff}} = s_1 s_2 \rho$ is used for evaluation)
- Skewness direction $\in \{++, +-, --\}$
- Φ set $\in \{A, B\}$ where A is symmetric and B is asymmetric

🔥 Direction symmetry

Only one mixed-direction case $(--)$ is included. When Φ is asymmetric, $(+ -)$ and $(- +)$ need not be equivalent under variable relabeling. If the goal is a fully symmetric factorial design, include both mixed-direction settings.

3 2 Data loading and preprocessing

```
suppressPackageStartupMessages({  
  library(dplyr)  
  library(tidyr)  
  library(readr)
```

```

library(ggplot2)
library(stringr)
library(knitr)
})

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(
  rep      = file.path(RES_DIR, "summary_replications.csv"),
  cond    = file.path(RES_DIR, "summary_conditions.csv"),
  design  = file.path(DATA_DIR, "sim_conditions.rds")
)

if (!all(file.exists(unlist(files)))) {
  message("Missing required input files. Run the Study 2 pipeline and analysis script first.")
  if (knitr:::is_html_output() || knitr:::is_latex_output()) knitr:::knit_exit()
}

# --- design grid ---
design_raw <- readRDS(files$design)
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)

# --- replication summaries ---
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)
rep_raw <- rep_raw |>
  filter(model %in% c("NG", "EG"))

# labels and factor hygiene
model_labels <- c(NG = "Normal (NG)", EG = "Exponential (EG)")

prep_rep <- function(df) {

```

```

df |>
  mutate(
    Model    = factor(model_labels[model], levels = model_labels),
    T        = factor(T),
    VARset   = factor(VARset, levels = sort(unique(VARset))),
    direction = factor(direction, levels = c("++", "+-", "--"))
  )
}

rep_df <- prep_rep(rep_raw)

# --- compute rho_eff implied by mirroring (see callout) ---
dir_chr <- as.character(rep_df$direction)
dir_s1 <- ifelse(substr(dir_chr, 1, 1) %in% c("-", "-"), -1, 1)
dir_s2 <- ifelse(substr(dir_chr, 2, 2) %in% c("-", "-"), -1, 1)
rep_df <- rep_df |>
  mutate(
    rho_eff_val = rho * (dir_s1 * dir_s2),
    rho_eff = factor(rho_eff_val, levels = sort(unique(rho_eff_val)))
  )

# --- adjust truth/bias/coverage for rho only (truth for other params already correct) ---
eps <- 1e-8
rep_df <- rep_df |>
  mutate(
    truth_eff = if_else(param == "rho", rho_eff_val, truth),
    bias_eff  = post_mean - truth_eff,
    rel_bias_eff = if_else(abs(truth_eff) < eps, bias_eff, bias_eff / abs(truth_eff)),
    cover95_eff = (195 <= truth_eff) & (u95 >= truth_eff)
  )

```

4 3 MCMC diagnostics overview

We classify each fitted replication as:

- **Clean:** $\hat{R} \leq 1.01$ and no post-warmup divergences.
- **Problematic:** $\hat{R} > 1.01$ or at least one divergence.
- **Failed/Error:** non-OK status or missing diagnostics.

```

RHAT_THRESHOLD <- 1.01

rep_df <- rep_df |>
  mutate(
    n_div_clean = if_else(is.na(n_div), 0L, as.integer(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"))
  )

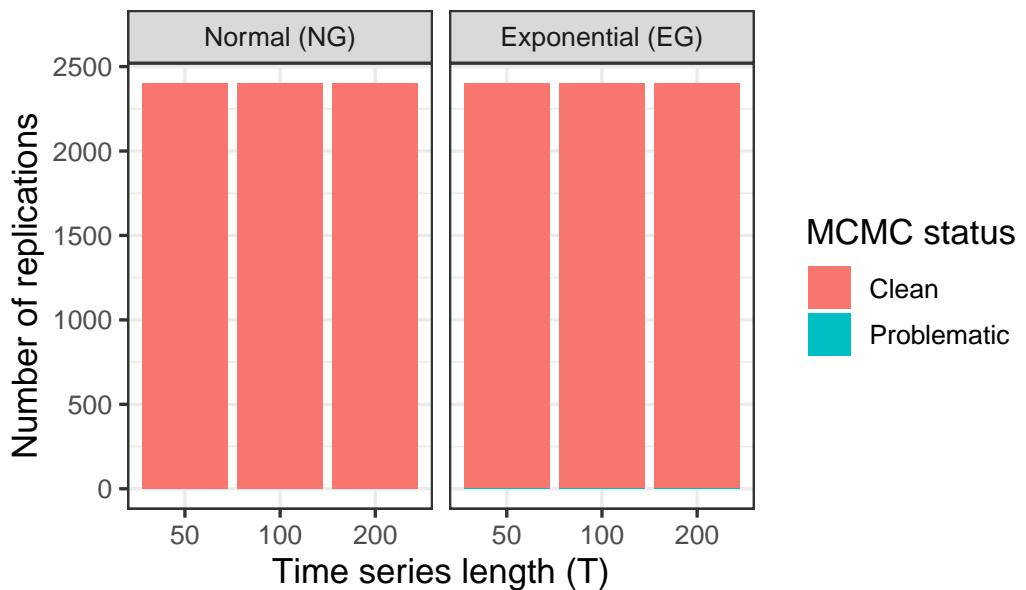
fit_level <- rep_df |>
  distinct(condition_id, rep_id, Model, T, mcmc_status, n_div_clean)

mcmc_summary <- fit_level |>
  group_by(Model, T, mcmc_status) |>
  summarise(Count = n(), .groups = "drop")

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 = "Fit status by model (Study 2: Exponential DGP)"
  ) +
  theme_bw(base_size = 13)

```

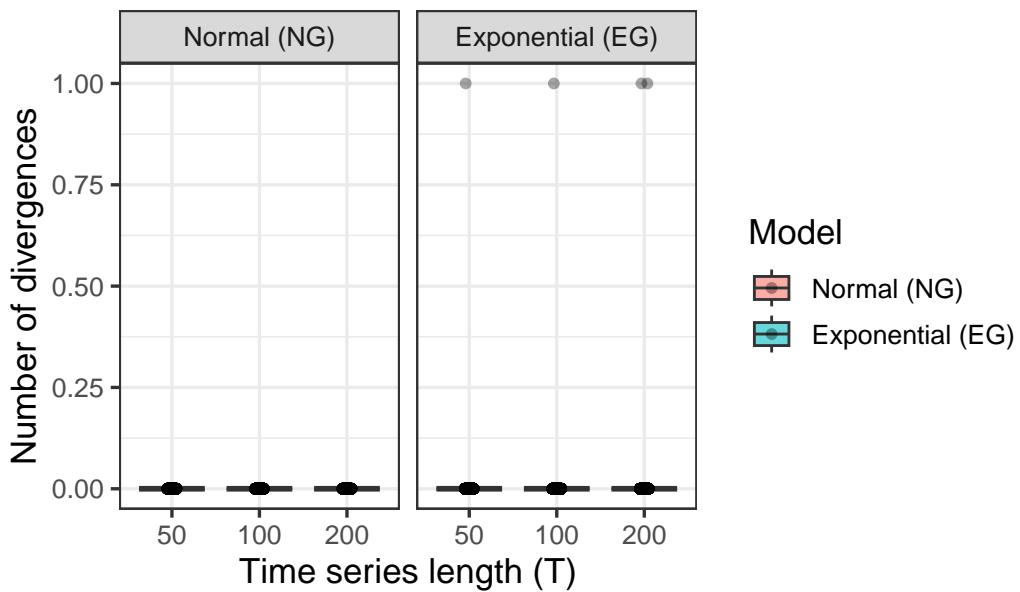
Fit status by model (Study 2: Exponential DGP)



```
div_dist <- fit_level |>
  filter(mcmc_status != "Failed/Error")

ggplot(div_dist, 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 = 1.4, alpha = 0.35,
    position = position_jitterdodge(jitter.width = 0.2, dodge.width = 0.8)
  ) +
  facet_wrap(~ Model) +
  theme_bw(base_size = 13) +
  labs(
    title = "Post-warmup divergence counts by model",
    y = "Number of divergences",
    x = "Time series length (T)"
  )
```

Post-warmup divergence counts by model



5 4 Core parameter recovery

5.1 4.1 Condition-level aggregation

All reported metrics below are computed from replication summaries using the adjusted ρ_{eff} truth for ρ .

```
core_params <- c("mu[1]", "mu[2]", "phi11", "phi12", "phi21", "phi22", "rho")  
  
summarise_conditions <- function(df) {  
  df |>  
    group_by(Model, T, direction, VARset, rho_eff, param) |>  
    summarise(  
      mean_rel_bias = mean(rel_bias_eff, na.rm = TRUE),  
      coverage_95 = mean(cover95_eff, na.rm = TRUE),  
      mean_post_sd = mean(post_sd, na.rm = TRUE),  
      emp_sd = sd(post_mean, na.rm = TRUE),  
      mean_bias = mean(bias_eff, na.rm = TRUE),  
      .groups = "drop"  
    ) |>  
    mutate(  
      emp_sd = if_else(is.na(emp_sd), 0, emp_sd),
```

```

        sd_bias = mean_post_sd - emp_sd,
        RMSE     = sqrt(mean_bias^2 + emp_sd^2)
    )
}

cond_core <- rep_df |>
  filter(param %in% core_params, mcmc_status != "Failed/Error") |>
  summarise_conditions()

```

5.2 4.2 Relative bias

```

theme_standard <- theme_bw(base_size = 13)
model_colors <- c("Normal (NG)" = "#377eb8", "Exponential (EG)" = "#4daf4a")

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

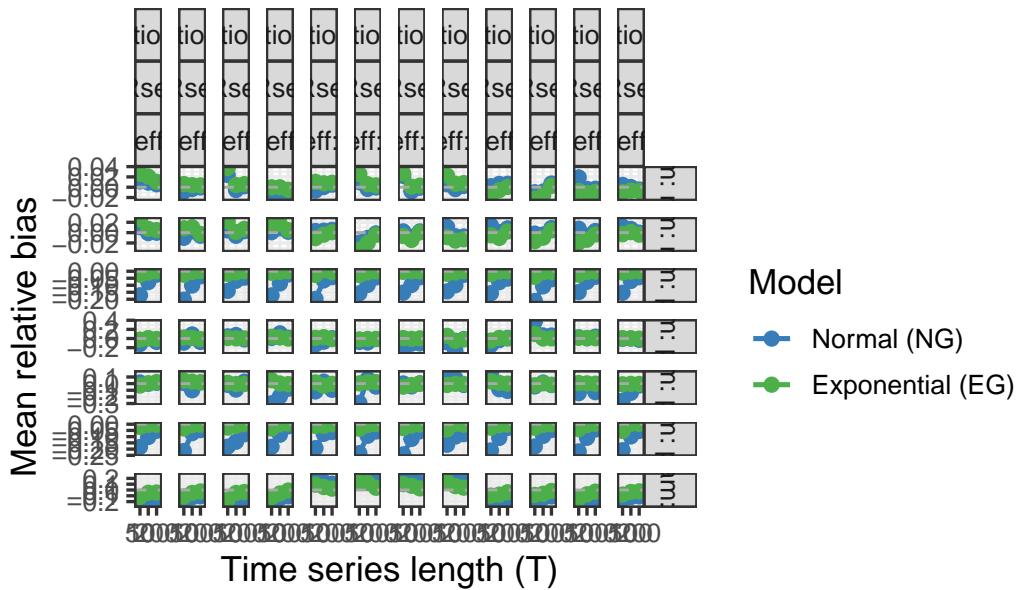
  p <- ggplot(data_filtered, aes(x = T, y = .data[[metric_col]], color = Model, group = Model))
  p <- p + geom_line(linewidth = 0.9) +
    geom_point(size = 2.2) +
    facet_grid(param ~ direction + VARset + rho_eff, labeller = label_both,
               scales = ifelse(use_free_y, "free_y", "fixed")) +
    theme_standard +
    scale_color_manual(values = model_colors) +
    labs(title = title, y = ylab, x = "Time series length (T)")

  if (!is.null(yref)) p <- p + geom_hline(yintercept = yref, linetype = "dashed", color = "darkblue")
  if (!is.null(ylims)) p <- p + coord_cartesian(ylim = ylims)
  p
}

plot_metric(cond_core, "mean_rel_bias", "Mean relative bias",
            "Relative bias (Study 2: Exponential DGP)", yref = 0, use_free_y = TRUE)

```

Relative bias (Study 2: Exponential DGP)

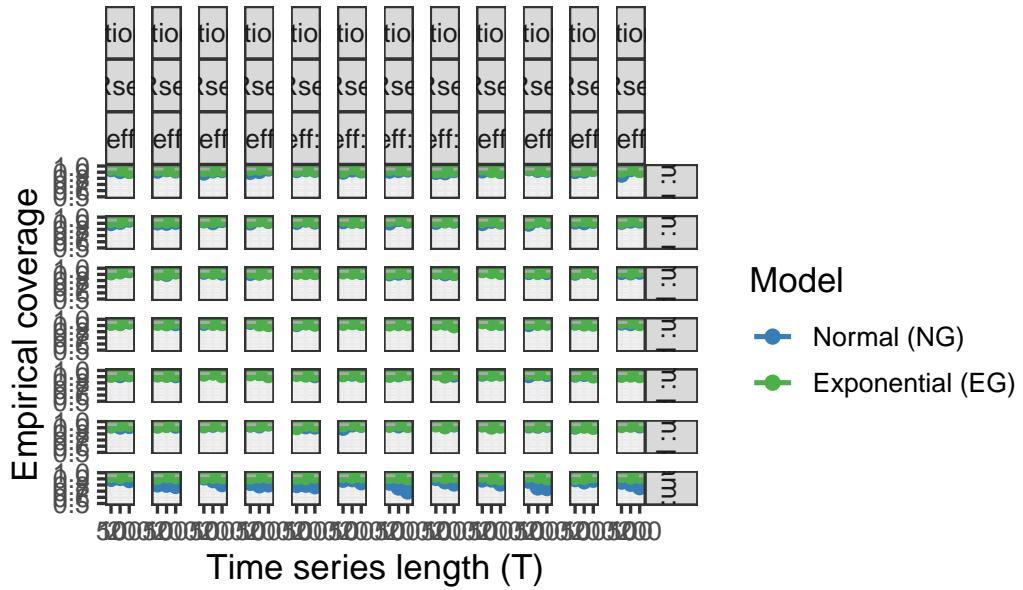


Interpretation. Across T , EG is approximately unbiased for Φ and μ . For ρ , the relevant comparison is to ρ_{eff} (callout above): under same-direction skewness, EG tends to recover ρ with small bias; NG exhibits attenuation in $|\rho|$, most pronounced at small T .

5.3 4.3 95% coverage

```
plot_metric(cond_core, "coverage_95", "Empirical coverage",
            "95% interval coverage (Study 2: Exponential DGP)", yref = 0.95, ylims = c(0.5,
```

95% interval coverage (Study 2: Exponential DGP)

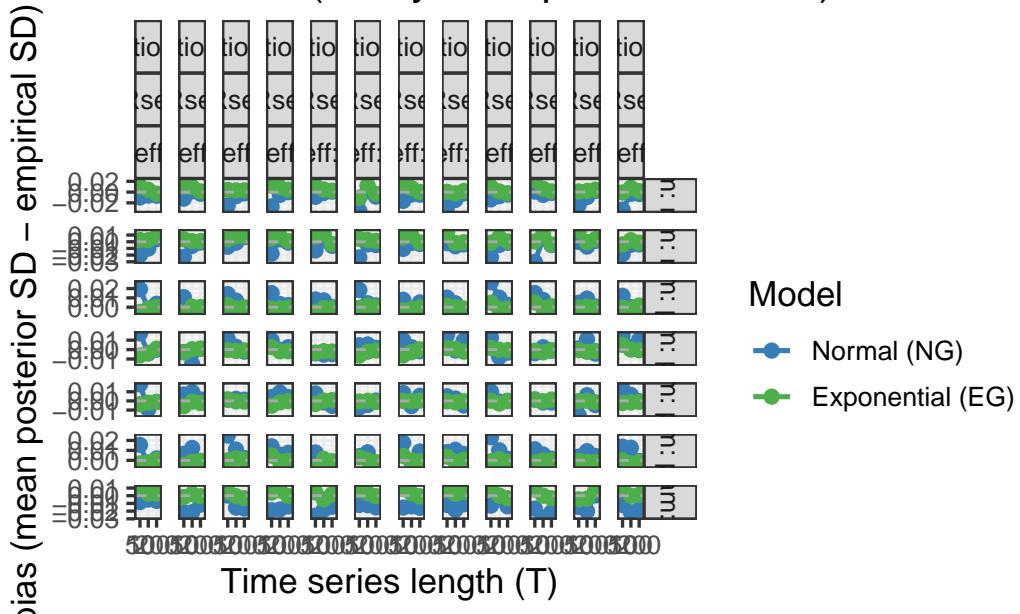


Interpretation. EG coverage is typically close to nominal for the core parameters. NG coverage shortfalls are concentrated in ρ , consistent with the attenuation bias under margin misspecification.

5.4 4.4 SD-bias

```
plot_metric(cond_core, "sd_bias", "SD-bias (mean posterior SD - empirical SD)",  
           "SD-bias (Study 2: Exponential DGP)", yref = 0, use_free_y = TRUE)
```

SD-bias (Study 2: Exponential DGP)



Interpretation. EG is generally well calibrated (SD-bias near 0). NG frequently shows negative SD-bias for ρ , indicating overconfident posterior uncertainty for dependence under misspecified margins.

6 5 Marginal scale parameters

6.1 5.1 EG: σ_{exp} (truth = 1)

```
cond_all <- rep_df |>
  filter(mcmc_status != "Failed/Error") |>
  summarise_conditions()

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

if (nrow(sigma_exp_data) > 0) {
  plot_metric(
    sigma_exp_data,
    metric_col = "mean_bias",
    ylab = "Mean bias (estimate - 1)",
    title = "EG: bias for sigma_exp (truth = 1)",
```

```

    yref = 0,
    use_free_y = TRUE
)
} else {
  message("No EG marginal scale parameters available.")
}

```

6.2 5.2 NG: σ (**truth** = 1)

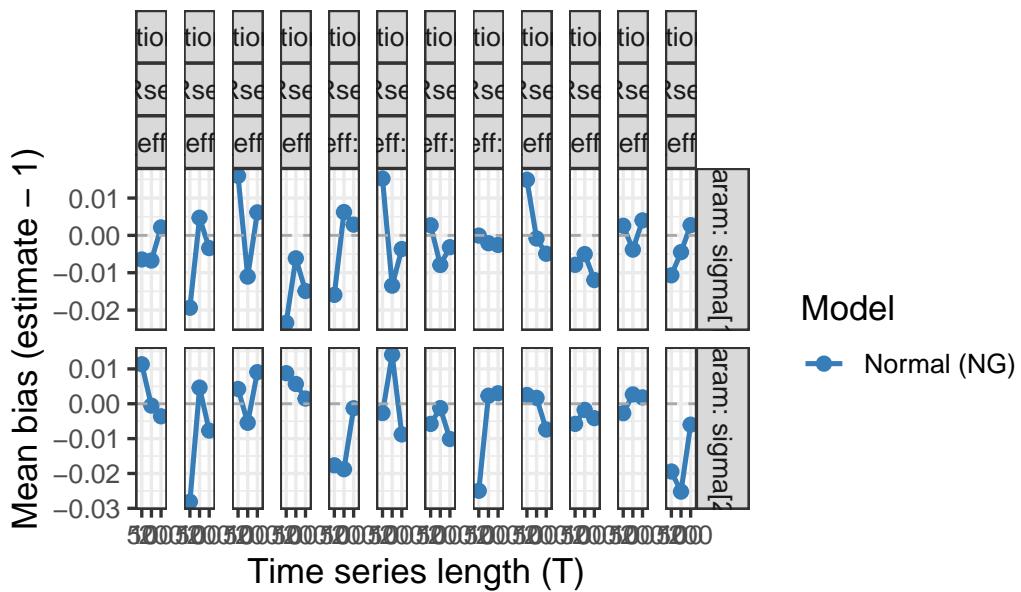
```

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

if (nrow(sigma_ng_data) > 0) {
  plot_metric(
    sigma_ng_data,
    metric_col = "mean_bias",
    ylab = "Mean bias (estimate - 1)",
    title = "NG: bias for sigma (truth = 1)",
    yref = 0,
    use_free_y = TRUE
)
} else {
  message("No NG marginal SD parameters available.")
}

```

NG: bias for sigma (truth = 1)



7 6 EG only: role of MCMC pathologies

7.1 6.1 Coverage split by MCMC status

```

cond_status <- rep_df |>
  filter(param %in% core_params, Model == "Exponential (EG)", mcmc_status != "Failed/Error")
  group_by(T, direction, VARset, rho_eff, param, mcmc_status) |>
  summarise(
    coverage_95 = mean(cover95_eff, na.rm = TRUE),
    .groups = "drop"
  )

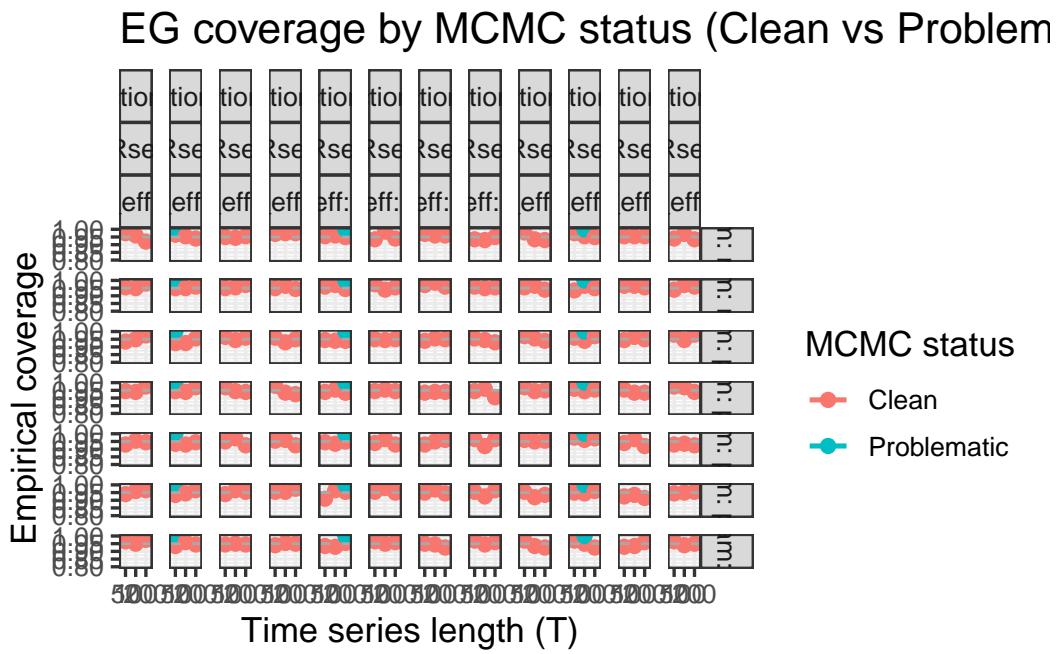
if (nrow(cond_status) > 0 && length(unique(cond_status$mcmc_status)) > 1) {
  ggplot(cond_status, aes(x = T, y = coverage_95, color = mcmc_status, group = mcmc_status))
    geom_line(linewidth = 0.9) +
    geom_point(size = 2.2) +
    geom_hline(yintercept = 0.95, linetype = "dashed", color = "darkgrey") +
    facet_grid(param ~ direction + VARset + rho_eff, labeller = label_both) +
    theme_standard +
    labs(
      title = "EG coverage by MCMC status (Clean vs Problematic)",

```

```

    y = "Empirical coverage",
    x = "Time series length (T)",
    color = "MCMC status"
) +
  coord_cartesian(ylim = c(0.8, 1.0))
} else {
  message("Insufficient EG status diversity for a split-by-status coverage plot.")
}

```



7.2 6.2 Absolute bias vs divergences

```

div_bias_data <- rep_df |>
  filter(Model == "Exponential (EG)", param %in% core_params, mcmc_status != "Failed/Error")
  mutate(abs_bias = abs(bias_eff)) |>
  distinct(condition_id, rep_id, T, direction, VARset, rho_eff, param, n_div_clean, abs_bias)

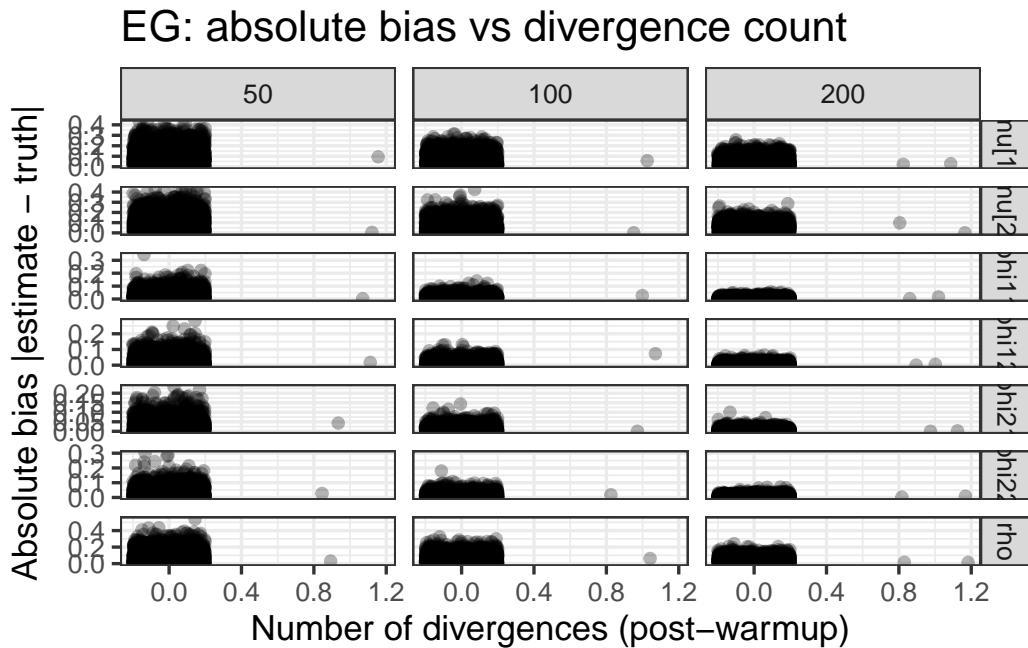
if (nrow(div_bias_data) > 0) {
  ggplot(div_bias_data, aes(x = n_div_clean, y = abs_bias)) +
    geom_point(alpha = 0.3, position = position_jitter(width = 0.2), size = 1.7) +
    geom_smooth(method = "gam", linewidth = 1.0) +
    facet_grid(param ~ T, scales = "free_y") +

```

```

theme_standard +
  labs(
    title = "EG: absolute bias vs divergence count",
    x = "Number of divergences (post-warmup)",
    y = "Absolute bias |estimate - truth|"
  )
}

```



8 7 Export tables

```

write_csv(cond_core, file.path(EXPORT_DIR, "study2_core_condition_summary.csv"))
write_csv(cond_all,  file.path(EXPORT_DIR, "study2_all_condition_summary.csv"))

```

9 8 Abstract

Under a bivariate VAR(1) with standardized Exponential innovations and Gaussian-copula dependence, we compare a Normal–Gaussian copula VAR (NG) with an Exponential–Gaussian copula VAR (EG). EG is generally well calibrated for μ and Φ and, when evaluated against the effective copula correlation $\rho_{\text{eff}} = s_1 s_2 \rho$ implied by margin mirroring in mixed-direction

designs, also recovers copula dependence with near-nominal interval coverage. NG is computationally stable but exhibits attenuation in dependence and under-coverage for ρ , consistent with misspecified marginal PITs. EG displays more divergent transitions at small sample sizes; nevertheless, within this design and tuning, coverage is typically similar across “Clean” and “Problematic” EG fits and absolute bias is only weakly related to divergence counts.