Study 1: Comparative Performance of Gaussian and Skew-Normal Copula VAR Models Under Marginal Misspecification

Table of contents

0. Summary

0.1 Computational Stability versus Statistical Inference

The Normal–Gaussian (NG) model is computationally stable across all simulation conditions (no post-warmup divergent transitions; max $\widehat{R} \leq 1.01$ in all replications). The Skew–Gaussian (SG) model becomes progressively harder to sample as marginal skewness increases: the proportion of replications classified as *Problematic* (max $\widehat{R} > 1.01$ and/or at least one divergent transition) rises under `extremeCHI`.

0.2 Model Performance Across Skewness Conditions

Moderate skewness (`moderateSN`, $\alpha = \pm 4$): NG and SG exhibit comparable performance. For $T \geq 100$, bias is near zero and coverage is close to 0.95 for all core parameters. At $T = 50$, both models show mild finite sample attenuation in some Φ and ρ parameters (due to priors, estimation uncertainty or errors in variables effect?), but the magnitudes are small and do not affect coverage. Under this level of skewness, SG's additional marginal flexibility yields no inferential benefit.

Strong skewness (`strongSN`, $\alpha = \pm 9$): Differences between NG and SG emerge for the VAR dynamics. SG typically reduces relative bias in Φ by approximately 0.05–0.15, and converges faster to the true values as T increases. Both models exhibit appreciable downward bias in ρ. For the NG model, this arises from marginal misspecification (assuming normal when the truth is skew-normal). For the SG model, despite being correctly specified in terms of the distributional family, the shape parameter α is difficult to estimate at small T (see Section 7.2): when $\hat{\alpha}$ is biased toward zero, the fitted CDF still deviates from the truth, producing PIT distortion and attenuated ρ estimates. As T increases and α becomes better identified, this distortion diminishes.

Extreme misspecification (`extremeCHI`, standardized χ^2_1 innovations): Performance deteriorates markedly, with parameter-specific failure modes: - VAR dynamics (Φ): SG remains comparatively robust—coverage is typically near nominal for $T \geq 100$—whereas NG shows substantial bias and undercoverage. - Copula correlation (ρ): both models exhibit severe attenuation; under mixed-sign conditions posterior means can cross zero, producing relative bias below -1. - Intercepts (μ): SG shows systematic bias in μ (absolute bias up to ≈ 0.2 in some settings), while NG's μ remains close to 0. This appears to reflect an interaction between the centered skew-normal parameterization and the inability of the skew-normal family to approximate the χ_1^2 tail behavior.

0.3 Insights

PIT distortion induced by marginal CDF misspecification is the dominant failure mechanism. In both models, misspecified marginals yield non-uniform PITs, which attenuate the effective dependence seen by the Gaussian copula and can induce apparent sign reversals under extreme mismatch. The NG model can partially accommodate skewness through small scale adjustments (typically $|\sigma - 1| \approx 0.02$–$0.03$), but this does not correct tail probabilities and therefore does not repair the PIT. Under `extremeCHI`, the SG model's intercept estimates shift away from 0, suggesting that the intercept absorbs part of the marginal misfit under the centered skew-normal parameterization; this empirical trade-off coincides with relatively stable estimation of Φ but does not remedy attenuation of ρ.

1. Introduction

This simulation study compares the performance of two Bayesian Vector Autoregressive (VAR(1)) models: a standard Normal-Gaussian (NG) model and a Skew-Gaussian (SG) model. The study investigates how these models recover the true parameters when the data generating process (DGP) exhibits varying degrees of skewness in the innovations, coupled by a Gaussian copula.

1.1. Data Generating Process (DGP)

The DGP is a bivariate VAR(1) model:

$$Y_t = \mu + \Phi Y_{t-1} + \epsilon_t$$

Where Y_t is a 2x1 vector of observations, μ is the intercept vector (set to 0 in this simulation), Φ is the 2x2 matrix of autoregressive coefficients, and ϵ_t is the 2x1 vector of innovations (errors).

The innovations $\epsilon_t = (\epsilon_{1,t}, \epsilon_{2,t})^T$ are generated such that they are standardized.

Note

Standardization of the innovations We require innovations $\epsilon_t = (\epsilon_{1,t}, \epsilon_{2,t})^\top$ to have $\mathbb{E}[\epsilon_{i,t}] = 0$ and $\mathrm{Var}(\epsilon_{i,t}) = 1$ for all conditions. This is enforced in the DGP, and the fitted models are parameterized to be consistent with this convention.

Skew-Normal marginals (DGP and SG model) Let $\mathrm{SN}(\xi, \omega, \alpha)$ denote the skew-normal with location ξ, scale $\omega > 0$, shape α.

$$\delta \;=\; \frac{\alpha}{\sqrt{1+\alpha^2}}$$

Moments:

$$\mu_{\mathrm{SN}} = \xi + \omega\,\delta\,\sqrt{\tfrac{2}{\pi}}, \qquad \sigma^2_{\mathrm{SN}} = \omega^2\Big(1 - \tfrac{2\delta^2}{\pi}\Big)$$

To impose mean 0 and variance 1, choose

$$\omega \;=\; \left(1 - \frac{2\delta^2}{\pi}\right)^{-\frac{1}{2}}, \qquad \xi \;=\; -\omega\,\delta\,\sqrt{\tfrac{2}{\pi}}$$

In the DGP (for `moderateSN` and `strongSN`), we set $\alpha \in \{\pm 4, \pm 9\}$, compute δ, then pick ω, ξ via the formulas above so that draws from $\mathrm{SN}(\xi, \omega, \alpha)$ are standardized.

In the SG model, we use centered parameterization that re-expresses the SN so that the innovation has zero theoretical mean by construction. In practice, the Stan parameters are (δ, ω) (or equivalent reparameterizations), and the implied ξ is that value which centers the innovation at (0) (as above). This is why the SG model cannot use μ to absorb a nonzero innovation mean; μ therefore represents the VAR intercept (process mean) rather than a marginal location parameter.

Chi-squared marginals (DGP and NG/SG models) For χ^2_ν: $\mathbb{E} = \nu$, $\mathrm{Var} = 2\nu$, skewness $\gamma_1 = \sqrt{\frac{8}{\nu}}$, excess kurtosis $\gamma_2 = \frac{12}{\nu}$. In the DGP (for `extremeCHI`), we standardize after simulating: if $X \sim \chi^2_1$, set

$$Z = \frac{X-1}{\sqrt{2}}, \qquad \text{and (optionally) mirror as } -Z \text{ for left skew.}$$

Thus Z has mean 0 and variance 1 but remains skewed ($\gamma_1 \approx 2.83$) and heavy-tailed ($\gamma_2 = 12$). In the NG model, innovations are Gaussian with scale σ. Because the data are standardized, the truth is $\sigma = 1$; deviations from 1 arise from misspecification.

In the SG model, innovations remain skew-normal with zero-mean by construction, using (δ, ω). Under a χ^2 DGP this implies misspecified marginals; the parameter α adjusts shape flexibly but cannot match χ^2_1 skewness.

Note

SG Model Parameterization: Variance Not Fixed
The SG model estimates both scale (ω) and shape (δ) parameters freely, rather than constraining the innovation variance to equal 1. This apparent overparameterization is acceptable for several reasons:

1. **Bayesian identification**: In a Bayesian framework with proper priors on both ω and δ, the posterior remains well-defined even without fixing variance. The data inform both parameters jointly, and the priors regularize the solution.

2. **Practical flexibility**: Fixing $\omega = f(\delta)$ to enforce unit variance would reduce model flexibility and could cause numerical issues when $|\delta| \to 1$ (where $\omega \to \infty$). The free parameterization avoids these boundary complications.

3. **Prior-induced regularization**: The half-normal prior on ω (centered near 1) and the normal prior on δ (centered at 0) together induce a prior on innovation variance that concentrates around reasonable values without hard constraints.

4. **Robustness to misspecification**: When the true DGP is not skew-normal (e.g., `extremeCHI`), the model can adjust both shape and scale to best approximate the data, rather than being forced into a potentially poor fit by the unit-variance constraint.

The joint distribution of ϵ_t is modeled using a Gaussian Copula, parameterized by the correlation coefficient ρ. This allows the dependence structure to be modeled independently of the marginal distributions $f_i(\epsilon_{i,t})$. The joint density is given by Sklar's theorem:

$$f(\epsilon_{1,t}, \epsilon_{2,t}) = c(u_{1,t}, u_{2,t}; \rho) \cdot f_1(\epsilon_{1,t}) \cdot f_2(\epsilon_{2,t})$$

$$f(\epsilon_{1,t}, \epsilon_{2,t}) = \underbrace{c_{\text{Gauss}}(F_1(\epsilon_{1,t}), F_2(\epsilon_{2,t}); \rho)}_{\text{Dependence Structure}} \cdot \underbrace{\frac{2}{\omega_1}\phi\left(\frac{\epsilon_{1,t} - \xi_1}{\omega_1}\right)\Phi\left(\alpha_1\frac{\epsilon_{1,t} - \xi_1}{\omega_1}\right)}_{\text{Marginal Density for } \epsilon_1} \cdot \underbrace{\frac{2}{\omega_2}\phi\left(\frac{\epsilon_{2,t} - \xi_2}{\omega_2}\right)\Phi\left(\alpha_2\frac{\epsilon_{2,t} - \xi_2}{\omega_2}\right)}_{\text{Marginal Density for } \epsilon_2}$$

The marginal distributions f_i are varied to introduce different levels and types of skewness.

Note

Numerical Stability in Copula Evaluation
The Gaussian copula density requires evaluating $\Phi^{-1}(u)$ where $u = F(\epsilon)$ is the probability integral transform. When u approaches 0 or 1, $\Phi^{-1}(u)$ diverges to $\pm\infty$, causing numerical overflow. To prevent this, we apply boundary clamping:

$$u_{\text{clamped}} = \max(\varepsilon, \min(1 - \varepsilon, u)), \quad \varepsilon = 10^{-9}$$

This clamping affects only the most extreme quantiles (beyond the 0.9999999th percentile) and has negligible impact on inference. The choice of $\varepsilon = 10^{-9}$ balances numerical stability against loss of tail information—values smaller than 10^{-12} risk floating-point underflow, while values larger than 10^{-6} would noticeably compress the effective copula support.

1.2. Simulation Design

The study employs a full factorial design crossing five factors, resulting in 108 unique conditions, with 200 replications per condition.

Table 1: Summary of the Simulation Design Factors.

Factor	Levels
Time Series Length (T)	50, 100, 200
Copula Correlation (ρ)	0.30, 0.50
VAR Parameters (Φ)	**Set A** (Symmetric): $\begin{pmatrix} 0.40 & 0.10 \\ 0.10 & 0.40 \end{pmatrix}$
	Set B (Asymmetric): $\begin{pmatrix} 0.55 & 0.10 \\ 0.10 & 0.25 \end{pmatrix}$
Skewness Level (Marginals)	`moderateSN`: Skew-Normal (SN), shape $\alpha = \pm 4$
	`strongSN`: SN, shape $\alpha = \pm 9$
	`extremeCHI`: Standardized Chi-squared (χ^2_1)
Skewness Direction	`++` (both positive), `--` (both negative), `+-` (mixed; `-+` omitted)

Note

Exclusion of -+ Direction
The design includes directions `++`, `--`, and `+-`, and omits `-+` to reduce computational cost.

- For **VAR Set A** ($\phi_{11} = \phi_{22}$ and $\phi_{12} = \phi_{21}$), relabeling $Y_1 \leftrightarrow Y_2$ leaves the DGP invariant. Under this symmetry, `+-` and `-+` are exactly equivalent, and including both would be redundant.
- For **VAR Set B**, the cross-effects remain symmetric ($\phi_{12} = \phi_{21}$), but the diagonal dynamics differ ($\phi_{11} \neq \phi_{22}$). Swapping Y_1 and Y_2 therefore also swaps ϕ_{11} and ϕ_{22}; `-+` is not strictly identical to `+-` *within the same VAR-Set-B condition.*

Accordingly, mixed-direction results for Set B should be interpreted qualitatively (e.g., patterns in PIT distortion and ρ attenuation) rather than as an exact surrogate for the omitted `-+` case.

Recommendations. 1. If exact exchangeability under mixed skewness is required, restrict mixed-direction analyses to VAR Set A. 2. If VAR Set B is substantively important, include the `-+` direction (and, if needed, the swapped-diagonal analogue of Set B) as a sensitivity analysis.

1.3. True Parameter Values

Table 2: True Parameter Values Used in the Data Generating Process.

Parameter	True Value	Notes
μ_1, μ_2	0, 0	Innovations are mean-zero
ϕ_{11} (Set A / Set B)	0.40 / 0.55	Diagonal AR coefficients
$\phi_{12} = \phi_{21}$	0.10	Cross-effects (symmetric)
ϕ_{22} (Set A / Set B)	0.40 / 0.25	Diagonal AR coefficients
ρ	0.30 or 0.50	Copula correlation
σ_1, σ_2 (NG model)	1.0, 1.0	Innovations are unit-variance
α (`moderateSN`)	±4 (direction-dependent)	$\delta \approx \pm 0.970$
α (`strongSN`)	±9 (direction-dependent)	$\delta \approx \pm 0.994$
ω (`moderateSN`)	1.5795	Derived from $\alpha = \pm 4$
ω (`strongSN`)	1.6415	Derived from $\alpha = \pm 9$
α, ω (`extremeCHI`)	Not applicable (misspecified)	SG parameters have no true counterpart

Note

Interpreting SG Parameters Under `extremeCHI`

For the `extremeCHI` condition, the DGP uses standardized χ^2_1 marginals, not skew-normal. Therefore, the SG model parameters (α, ω) have no "true" values in the conventional sense—any estimate represents the model's best skew-normal approximation to a non-skew-normal distribution.

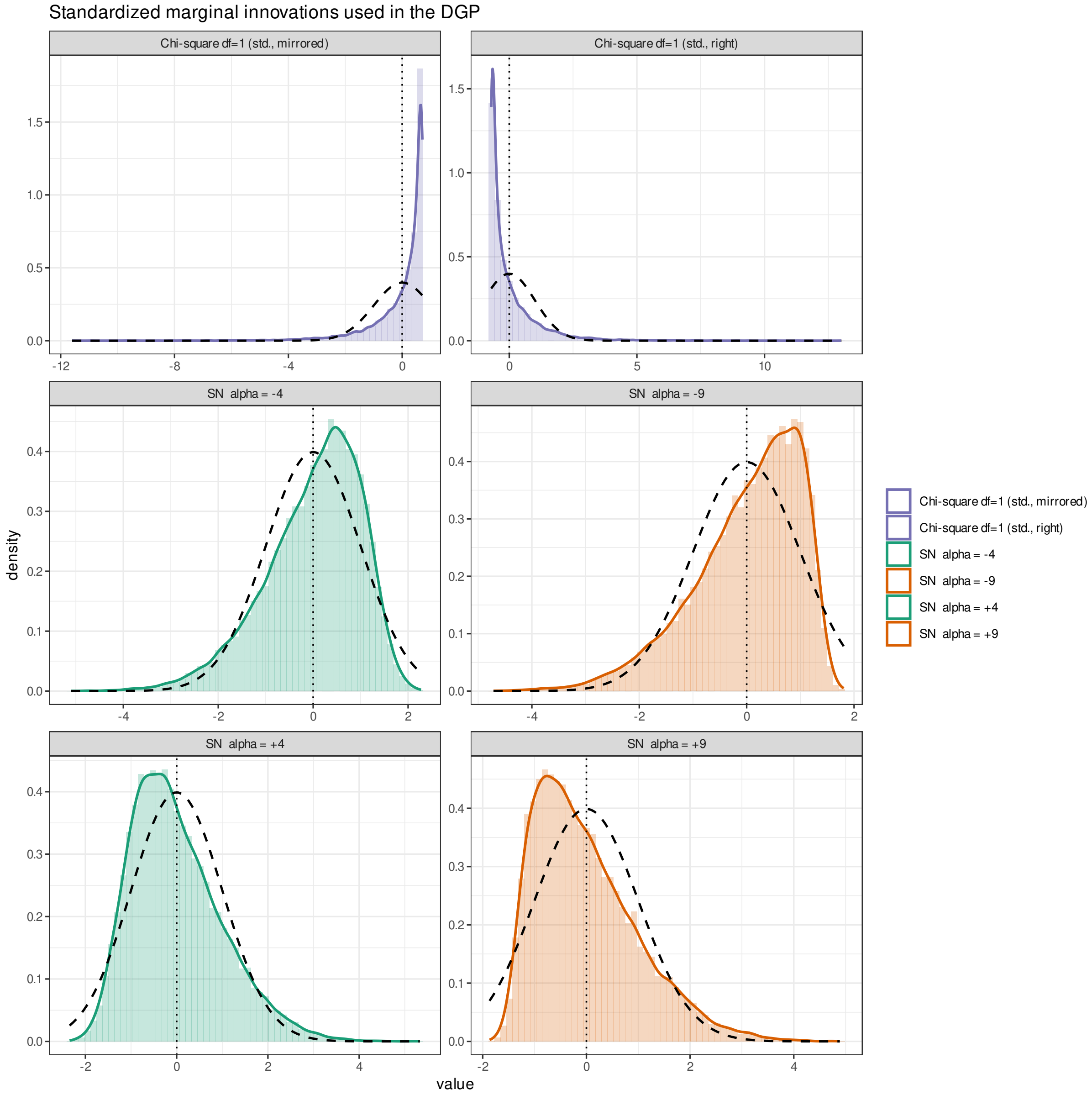
In the analysis, we set these truth values to `NA` and exclude them from bias/coverage calculations. When interpreting SG model fits under `extremeCHI`, the estimated α and ω should be understood as:

- α: The skewness direction and magnitude that best approximates χ^2_1 within the skew-normal family (expected to be large and positive for right-skewed conditions)
- ω: The scale adjustment needed to match variance given the fitted shape

These estimates reflect approximation quality rather than parameter recovery.

1.4 Visual check: standardized marginal innovations (DGP)

```r
# SN parameters that enforce mean 0 and var 1 for a given alpha
sn_params <- function(alpha) {
  delta <- alpha / sqrt(1 + alpha^2)
  omega <- sqrt(1 / (1 - 2 * delta^2 / pi))       # Var = 1
  xi    <- -omega * delta * sqrt(2 / pi)          # Mean = 0
  list(xi = xi, omega = omega, alpha = alpha)
}

set.seed(2025)

# standardized chi-square draws (df = 1), both right-skewed and mirrored
rchisq_std <- function(n, df = 1, mirror = FALSE) {
  x  <- stats::rchisq(n, df = df)
  z  <- (x - df) / sqrt(2 * df)  # mean 0, var 1
  if (mirror) -z else z
}

# 20k draws per margin
draws <- list(
  "SN  alpha = +4"              = sn::rsn(20000, xi = sn_params(+4)$xi, omega = sn_params(+4
  "SN  alpha = -4"              = sn::rsn(20000, xi = sn_params(-4)$xi, omega = sn_params(-4
  "SN  alpha = +9"              = sn::rsn(20000, xi = sn_params(+9)$xi, omega = sn_params(+9
  "SN  alpha = -9"              = sn::rsn(20000, xi = sn_params(-9)$xi, omega = sn_params(-9
  "Chi-square df=1 (std., right)"       = rchisq_std(20000, df = 1, mirror = FALSE),
  "Chi-square df=1 (std., mirrored)"    = rchisq_std(20000, df = 1, mirror = TRUE)
)

df_m <- dplyr::bind_rows(lapply(names(draws), function(nm) {
  tibble::tibble(value = draws[[nm]], dist = nm)
}))

# palette
pal <- c("SN  alpha = +4"           = "#1b9e77",
         "SN  alpha = -4"           = "#1b9e77",
         "SN  alpha = +9"           = "#d95f02",
         "SN  alpha = -9"           = "#d95f02",
         "Chi-square df=1 (std., right)"    = "#7570b3",
         "Chi-square df=1 (std., mirrored)" = "#7570b3")

ggplot(df_m, aes(value)) +
  geom_histogram(aes(y = after_stat(density), fill = dist),
                 bins = 60, alpha = 0.25, colour = NA) +
  geom_density(aes(colour = dist), linewidth = 0.8) +
  # N(0,1) reference
  stat_function(fun = dnorm, linewidth = 0.7, linetype = "dashed") +
  geom_vline(xintercept = 0, linetype = "dotted") +
  scale_fill_manual(values = pal, guide = "none") +
  scale_colour_manual(values = pal, name = "") +
  facet_wrap(~ dist, scales = "free", ncol = 2) +
  theme_bw(base_size = 10) +
  labs(title = "Standardized marginal innovations used in the DGP",
       x = "value", y = "density")
```



2. Data Loading and Preparation

```
# load the design grid
design <- readRDS(files$design) |>
  select(condition_id, skew_level, direction, T, rho, VARset)

# load condition-level summary (aggregated metrics)
```

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

# load replication-level summary (individual runs)
rep_raw <- read_csv(files$rep, show_col_types = FALSE) |>
  # NOTE: keep rows with param = NA so that failed fits (status != "ok") are
  # retained for MCMC-status summaries and replication-count tables.
  left_join(design, by = "condition_id")


# define parameter order and groups
param_levels <- c("omega[1]","omega[2]","alpha[1]","alpha[2]",
                  "sigma[1]","sigma[2]",
                  "mu[1]","mu[2]",
                  "phi11","phi12","phi21","phi22","rho")

# apply factor levels and clearer labels
prep_data <- function(df) {
  df |>
    mutate(param = factor(param, levels = param_levels),
           T = factor(T),
           skew_level = factor(skew_level, levels = c("moderateSN", "strongSN", "extremeCHI"
           rho_val = rho, # keep numeric rho
           VARset_val = VARset, # keep character VARset
           # MODIFIED: Removed manual prefixes (e.g., "VARset:").
           # We rely on ggplot's labeller=label_both used later.
           rho = factor(rho, labels = sort(unique(df$rho))),
           VARset = factor(VARset, labels = sort(unique(df$VARset))),
           Model = factor(ifelse(model == "SG", "SG", "NG"),
                          levels = c("NG", "SG")))
}

cond <- prep_data(cond_raw)
rep_df <- prep_data(rep_raw)


cond <- cond |>
  mutate(
    # use coalesce to handle NA emp_sd if N_valid < 2
    RMSE = sqrt(mean_bias^2 + coalesce(emp_sd^2, 0))
  )
```

```r
# separate core parameters (VAR dynamics, intercepts, correlation)
core_params <- c("mu[1]","mu[2]", "phi11","phi12","phi21","phi22","rho")
```

Note

Bias Metric for Intercepts (μ)
The "Relative Bias" plots display `mean_rel_bias`, defined as:

$$\text{Relative Bias} = \frac{\hat{\theta} - \theta_{\text{true}}}{|\theta_{\text{true}}|}$$

However, for the intercept parameters μ_1 and μ_2, the true value is zero, making relative bias undefined. In these cases, we report **absolute bias** instead:

$$\text{Bias}_{\mu} = \hat{\mu} - 0 = \hat{\mu}$$

This means that for μ panels in "Relative Bias" plots, the y-axis shows absolute deviation from zero (in original units), not a proportion. Cross-parameter comparisons should account for this difference in scale interpretation.

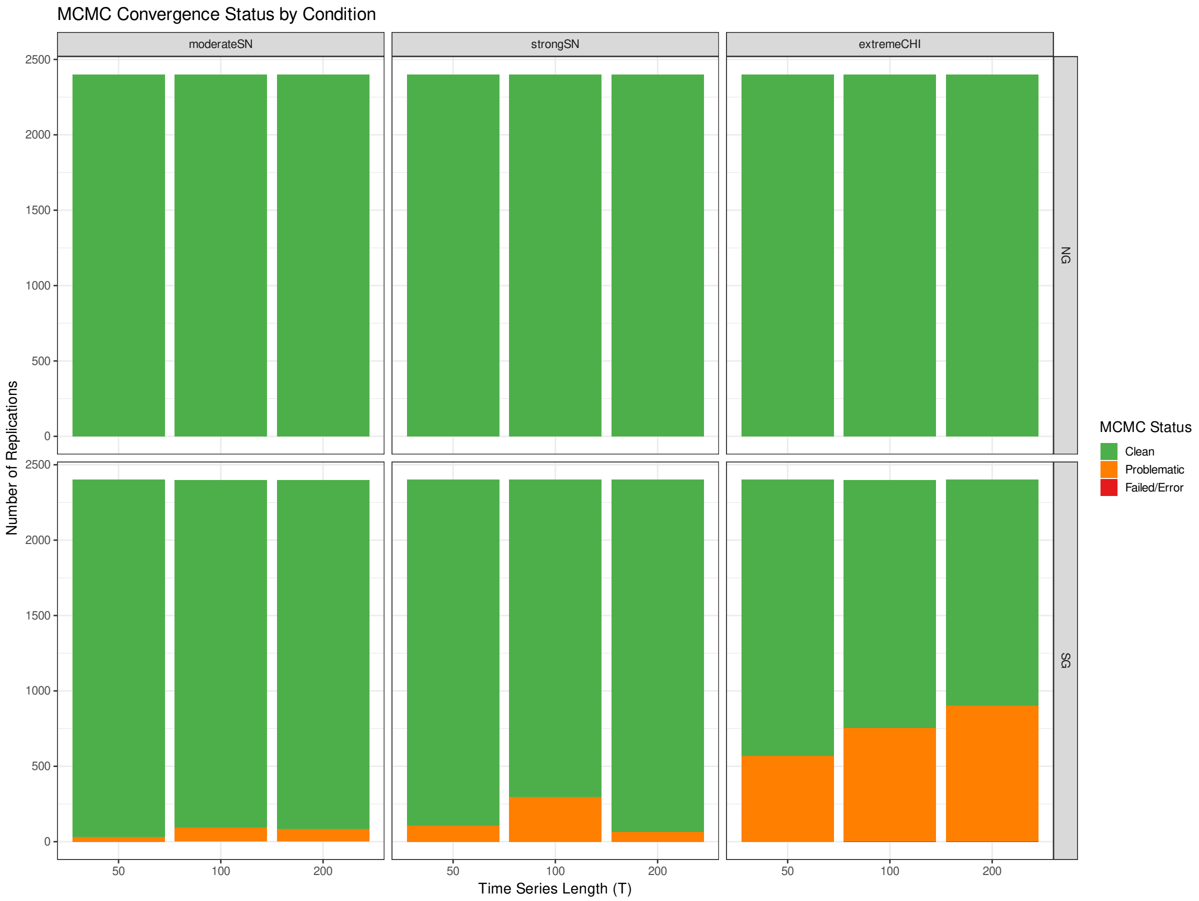
2.1. MCMC Classification and Overview

We classify runs based on MCMC diagnostics (R-hat and divergent transitions `n_div`) and summarize the computational performance. A replication run was deemed as "Problematic" if it successfully completed sampling but exhibited either of the following conditions:

- High R-hat: The potential scale reduction factor (`max_rhat`) for any parameter was greater than 1.01.
- Divergent Transitions: The sampler reported one or more divergent transitions (`n_div > 0`) after warmup.

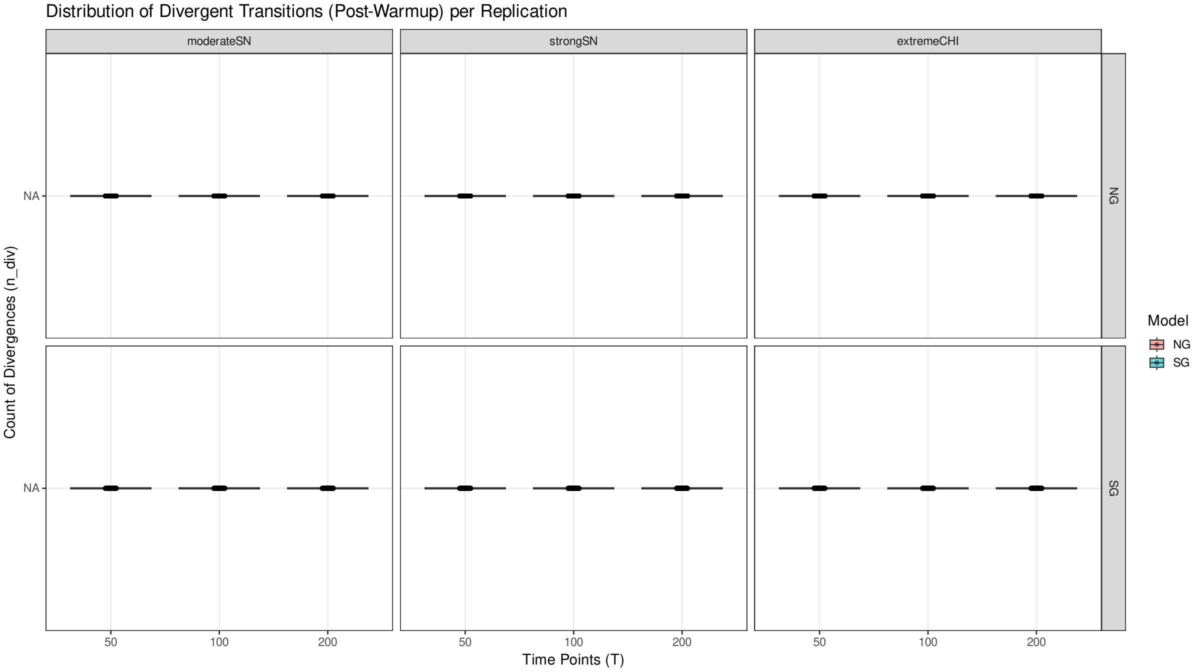
```r
RHAT_THRESHOLD <- 1.01

rep_df <- rep_df |>
  mutate(
    mcmc_status = case_when(
      is.na(max_rhat) | status != "ok" ~ "Failed/Error",
      # classification based on Rhat and divergent transitions (n_div)
      max_rhat > RHAT_THRESHOLD | n_div > 0 ~ "Problematic",
      TRUE ~ "Clean"
    ),
    mcmc_status = factor(mcmc_status, levels = c("Clean", "Problematic", "Failed/Error"))
  )

# MCMC Overview Plot (Status Counts)
mcmc_summary <- rep_df |>
  distinct(condition_id, rep_id, Model, mcmc_status, T, skew_level) |>
  group_by(Model, T, skew_level, mcmc_status) |>
  summarise(Count = n(), .groups = "drop")
```

```r
ggplot(mcmc_summary, aes(x = T, y = Count, fill = mcmc_status)) +
  geom_bar(stat = "identity", position = "stack") +
  facet_grid(Model ~ skew_level) +
  labs(x = "Time Series Length (T)", y = "Number of Replications", fill = "MCMC Status",
       title = "MCMC Convergence Status by Condition") +
  theme_bw(base_size = 14) +
  scale_fill_manual(values = c("Clean" = "#4daf4a", "Problematic" = "#ff7f00", "Failed/Error
```



Interpretation: NG fits are uniformly *Clean* across the design. SG sampling degrades with increasing marginal skewness: most runs are *Clean* under `moderateSN`, many runs are *Problematic* under `strongSN` (particularly at $T = 50$), and under `extremeCHI` almost all runs are *Problematic*.

```r
# MCMC Overview Plot (Distribution of Divergent Transitions)
# we use the replication level data (rep_df) to see the distribution of n_div across individu

div_dist_data <- rep_df |>
    # ensure we only look at the diagnostics once per replication run (n_div is the same for
    # we filter by one parameter (e.g., rho) to get unique runs, ensuring we have the n_div v
    filter(param == "rho") |>
    distinct(condition_id, rep_id, Model, T, skew_level, n_div, mcmc_status) |>
    # exclude runs that failed entirely before sampling (if any)
    filter(mcmc_status != "Failed/Error")

ggplot(div_dist_data, aes(x = T, y = n_div, fill = Model)) +
  # boxplot showing the distribution of divergences per condition
  # we hide default outliers as we will show them with jitter
  geom_boxplot(outlier.shape = NA, alpha = 0.6, position = position_dodge(width = 0.8)) +
  # jitter points to show individual runs; this helps visualize the density and outliers
  # position_jitterdodge allows jittering within the dodged groups (Models)
  geom_point(size = 1.5, alpha = 0.4, position = position_jitterdodge(jitter.width = 0.2, dod
  facet_grid(Model ~ skew_level) +
  theme_bw(base_size = 14) +
  # log1p scale (log(y+1)) might be better if the distribution is highly skewed with very lar
  # scale_y_continuous(trans = 'log1p', breaks = c(0, 5, 50, 200)) +
  labs(title = "Distribution of Divergent Transitions (Post-Warmup) per Replication",
       y = "Count of Divergences (n_div)",
       x = "Time Points (T)")
```



Interpretation: The divergence distributions corroborate the status counts. NG exhibits no divergent transitions. SG shows frequent divergences under `strongSN` and especially `extremeCHI`, with a heavy-tailed distribution of divergence counts: many replications have few divergences, but some exhibit dozens or more. Divergences diminish with increasing T in the SN DGPs but remain pervasive under `extremeCHI`.

3. Helpers

```
# standardized visualization settings
theme_standard <- theme_bw(base_size = 14) # to increase base font size
dodge_width <- 0.3

# helper function for plotting metrics across conditions
plot_metric <- function(data, metric_col, ylab, title, use_free_y = FALSE, ylims = NULL) {

  # filter out potential NAs (e.g. if N_truth_avail = 0)
  data_filtered <- data |> filter(!is.na(.data[[metric_col]]))

  if (nrow(data_filtered) == 0) {
      message("Skipping plot '", title, "' due to missing data.")
      return(NULL)
  }
```

```r
  p <- ggplot(data_filtered, aes(x = T, y = .data[[metric_col]], color = Model, group = Mode
    geom_line(position = position_dodge(dodge_width), linewidth = 1) +
    geom_point(position = position_dodge(dodge_width), size = 2.5) +
    # labeller = label_both correctly adds the variable name (VARset, rho) and the value.
    facet_grid(param ~ direction + VARset + rho, labeller = label_both, scales = ifelse(use_
    theme_standard +
    labs(title = title, y = ylab, x = "Time Points (T)")

  # add reference lines based on the metric
  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")
  }

  # apply custom Y-axis limits if provided
  if (!is.null(ylims)) {
    p <- p + coord_cartesian(ylim = ylims)
  }

  return(p)
}

# wrapper to filter data and call the plotting function for a specific skew level
generate_plots_for_condition <- function(skew_lvl) {
  data_subset <- cond |>
    filter(skew_level == skew_lvl, param %in% core_params)

  # Adjust Y-axis limits for coverage based on the condition severity
  cov_ylims <- if (skew_lvl == "extremeCHI") c(0.5, 1.0) else c(0.8, 1.0)

  list(
    # Metric: Relative Bias
    bias = plot_metric(data_subset, "mean_rel_bias", "Mean Relative Bias",
                       paste("Relative Bias (DGP:", skew_lvl, ")"), use_free_y = TRUE),
    # Metric: 95% CI Coverage
    coverage = plot_metric(data_subset, "coverage_95", "Empirical Coverage",
                           paste("95% Coverage (DGP:", skew_lvl, ")"), ylims = cov_ylims),
    # Metric: RMSE (Overall Accuracy)
    rmse = plot_metric(data_subset, "RMSE", "Root Mean Squared Error",
                       paste("RMSE (DGP:", skew_lvl, ")"), use_free_y = TRUE),
    # Metric: Posterior SD (Uncertainty Estimate)
    post_sd = plot_metric(data_subset, "mean_post_sd", "Mean Posterior SD",
                          paste("Mean Posterior SD (DGP:", skew_lvl, ")"), use_free_y = TRUE)
    # Metric: SD-Bias (Calibration of Uncertainty)
    sdbias = plot_metric(data_subset, "sd_bias", "SD-Bias",
                         paste("SD-Bias (DGP:", skew_lvl, ")"), use_free_y = TRUE)
  )
}
```

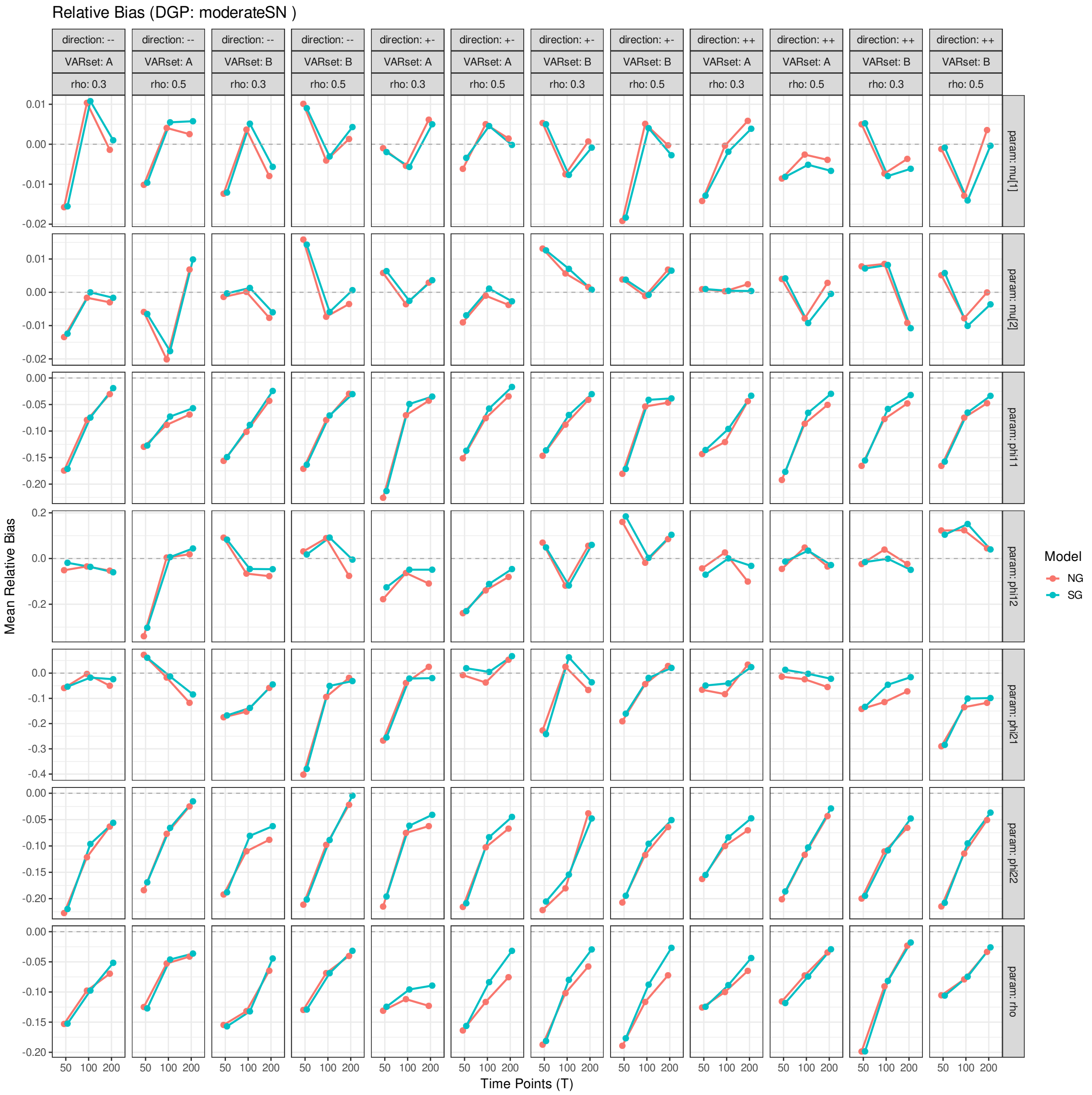
4. Condition 1: Moderate Skewness (moderateSN)

DGP: Skew-Normal innovations ($\alpha = 4$). NG model is misspecified; SG model is correctly specified.

```
plots_mod <- generate_plots_for_condition("moderateSN")
```

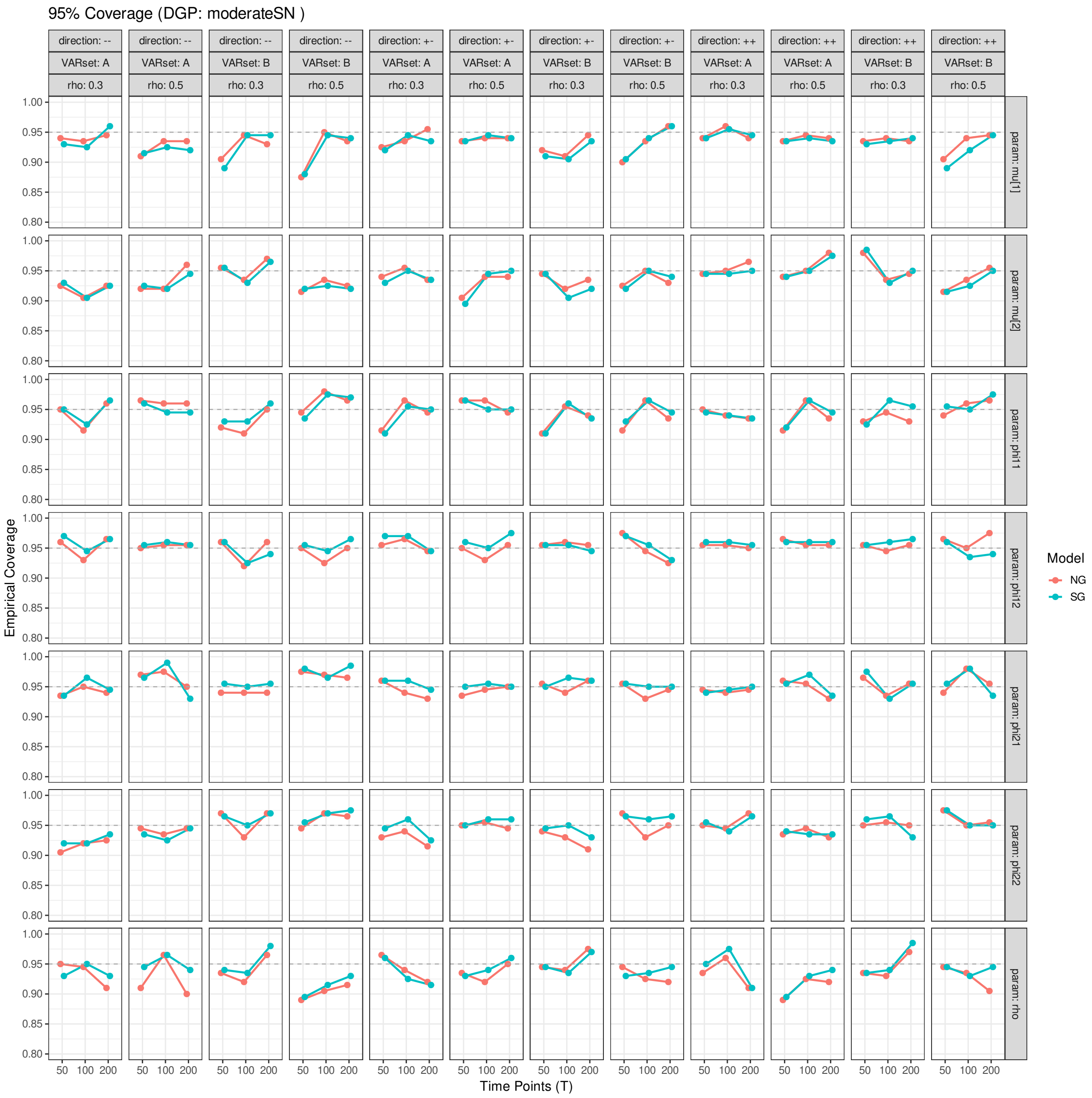
4.1. Relative Bias (moderateSN)

```
print(plots_mod$bias)
```



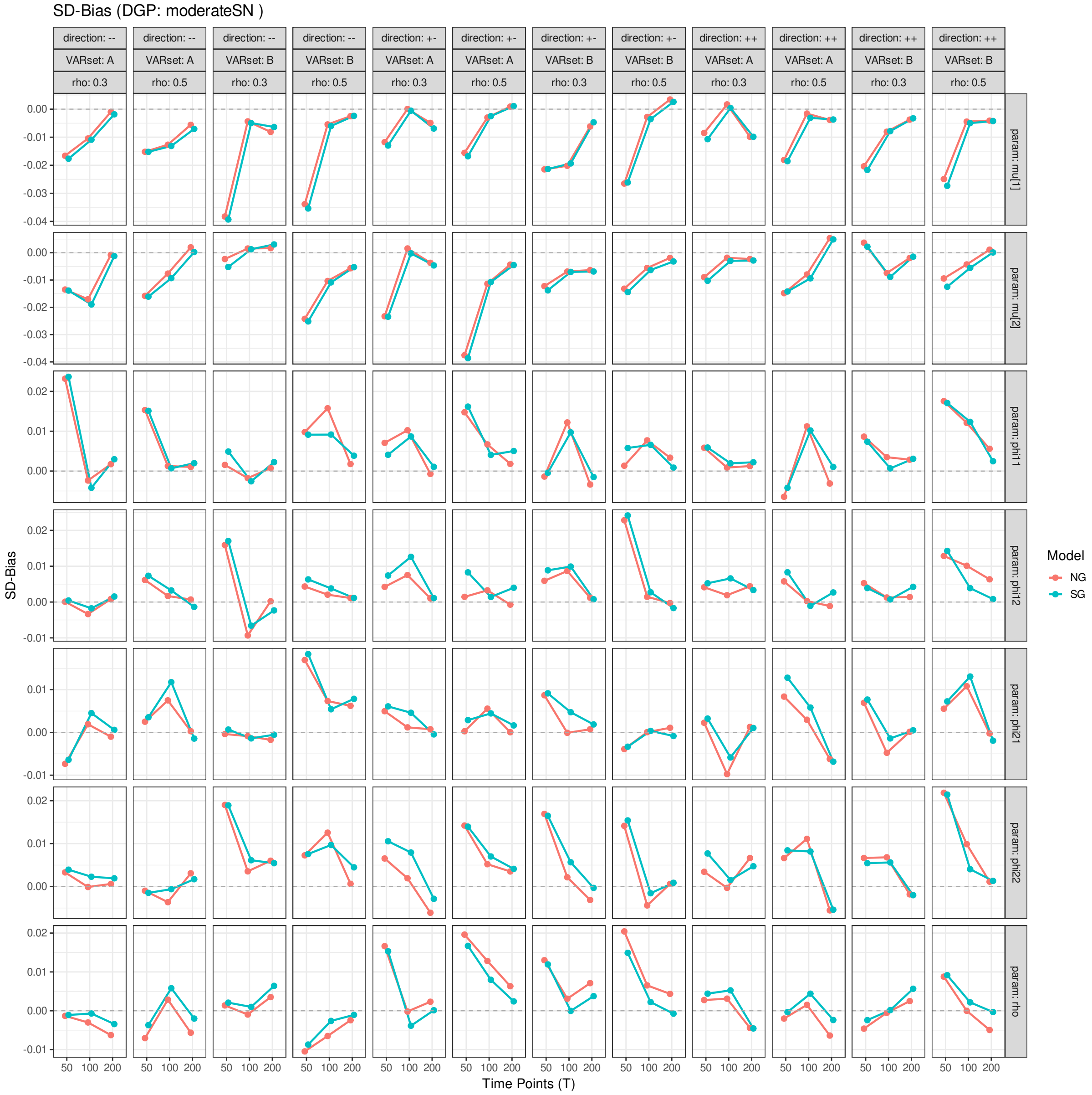
4.2. 95% Coverage (moderateSN)

```
print(plots_mod$coverage)
```



4.3. SD-Bias (moderateSN)

```
# Note: RMSE and Mean Posterior SD plots are omitted for brevity, focusing on Bias, Coverage
print(plots_mod$sdbias)
```



Interpretation (moderateSN): Under `moderateSN`, NG and SG have similar operating characteristics. Bias is small and coverage is near 0.95 for most parameters; modest finite-sample attenuation in some Φ and ρ panels is visible at $T = 50$ but largely disappears by $T \geq 100$. SD-bias is close to zero for both models, indicating reasonably calibrated posterior uncertainty. Under this level of skewness, SG's added complexity is not supported by material inferential gains.

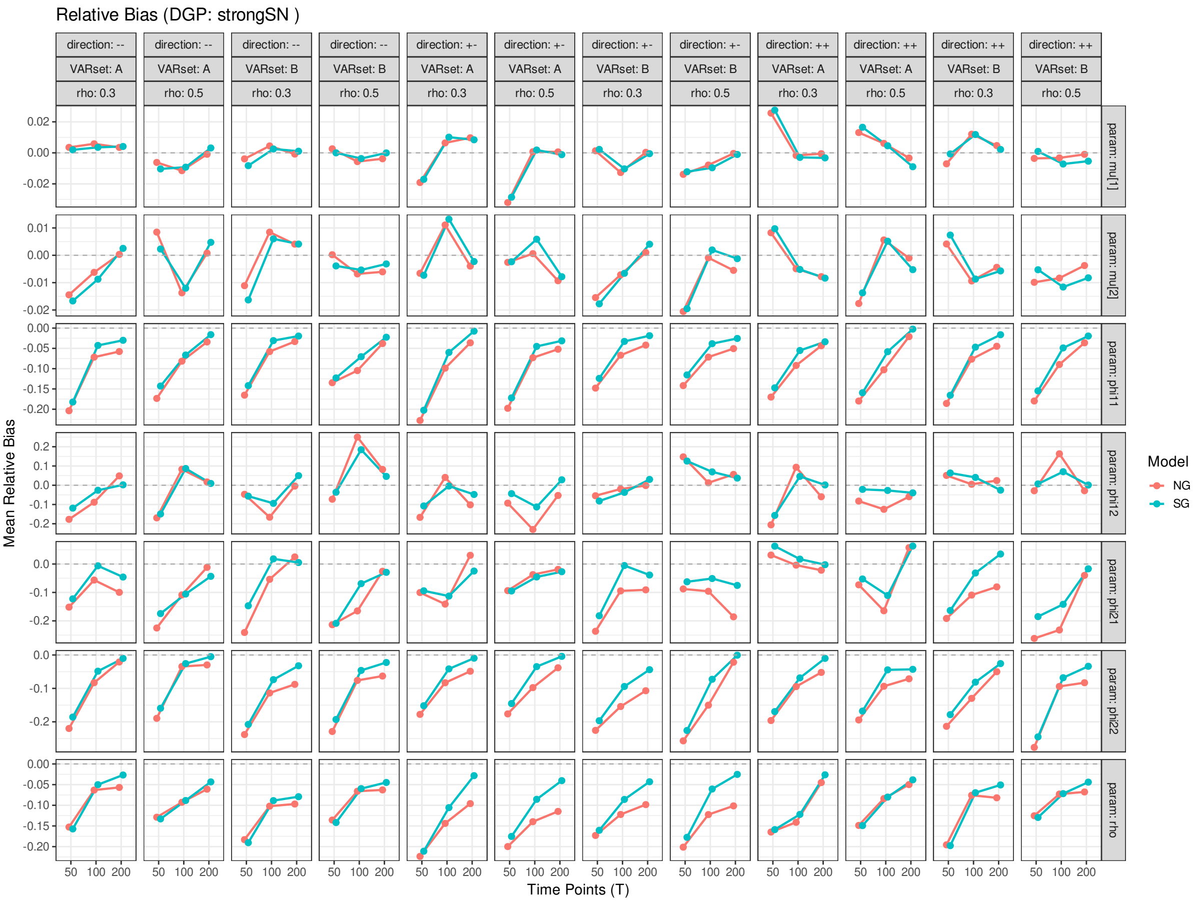
5. Condition 2: Strong Skewness (strongSN)

DGP: Skew-Normal innovations ($\alpha = 9$). The misspecification for the NG model is severe.

```r
plots_strong <- generate_plots_for_condition("strongSN")
```

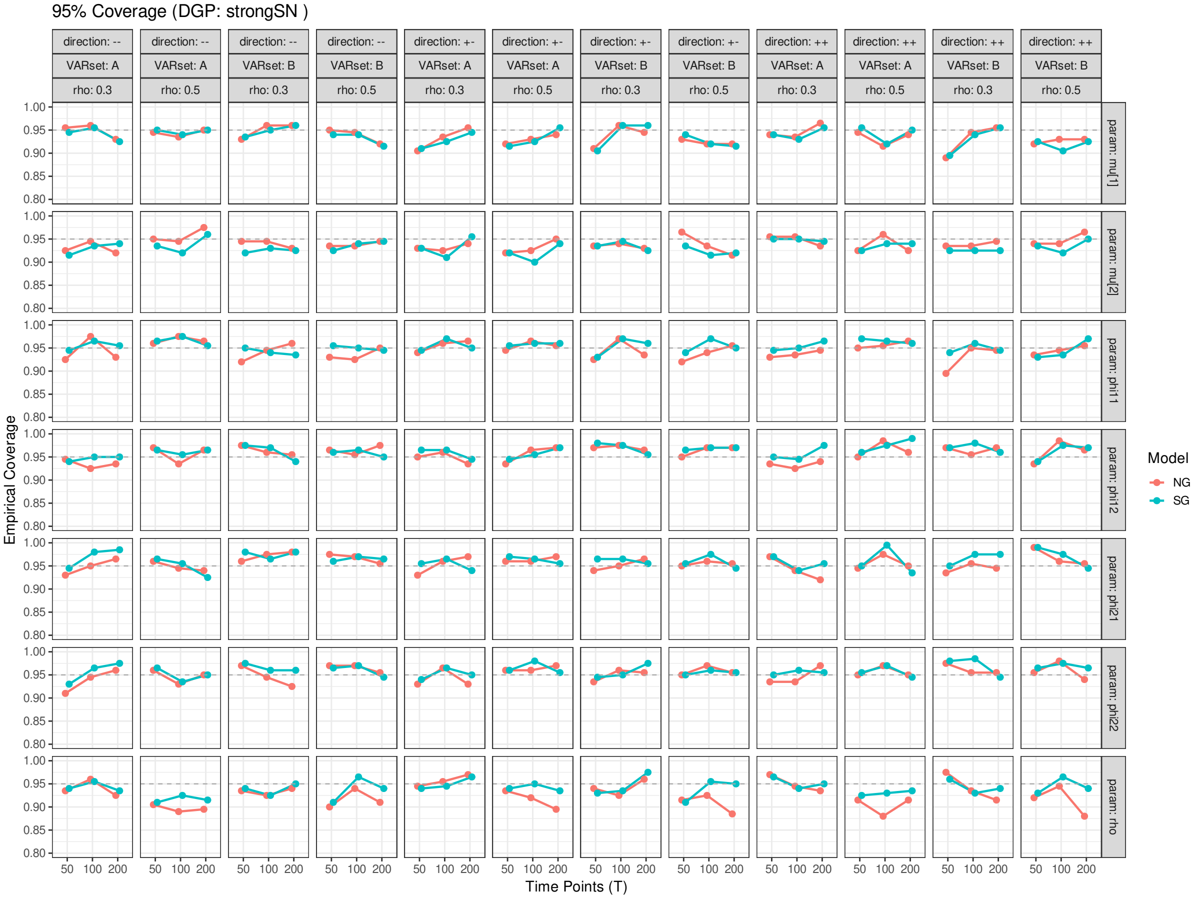
5.1. Relative Bias (strongSN)

```r
print(plots_strong$bias)
```



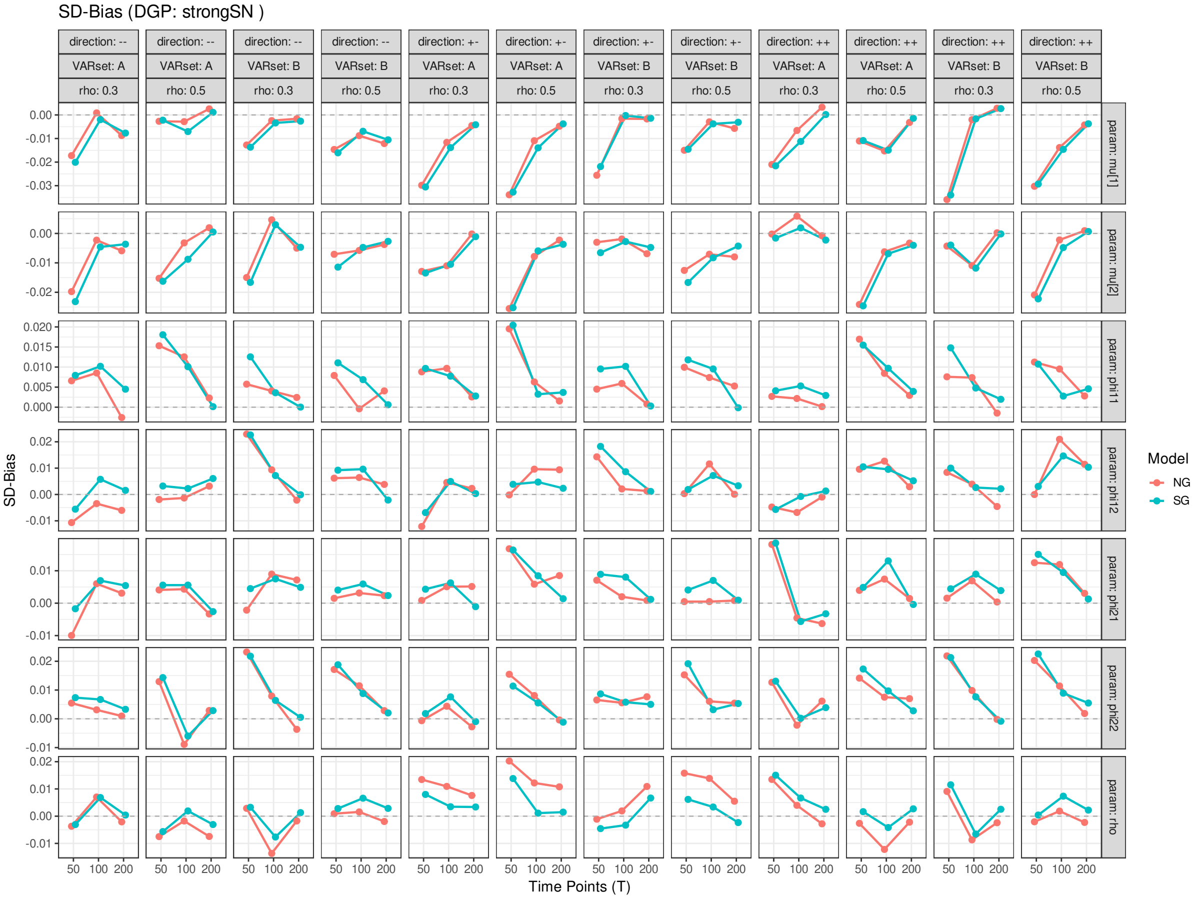
5.2. 95% Coverage (strongSN)

```
print(plots_strong$coverage)
```



5.3. SD-Bias (strongSN)

```
print(plots_strong$sdbias)
```



Interpretation (strongSN): Under strong skewness ($\alpha = \pm 9$), meaningful differences between the NG and SG models begin to emerge, though they remain modest. The SG model shows slightly reduced bias for the VAR dynamics (Φ), with typical improvements of 0.05–0.15 in relative bias compared to NG. Coverage rates for both models remain near nominal for most parameters, with the SG model showing marginally better calibration. The copula parameter ρ exhibits downward bias under both models, though this is less severe than under `extremeCHI`. SD-Bias remains small for both models, indicating that posterior uncertainty quantification is reasonably well-calibrated even under strong skewness. The computational cost of the SG model (increased divergences at $T = 50$) may not be justified by the modest improvements in inference, particularly for shorter time series.

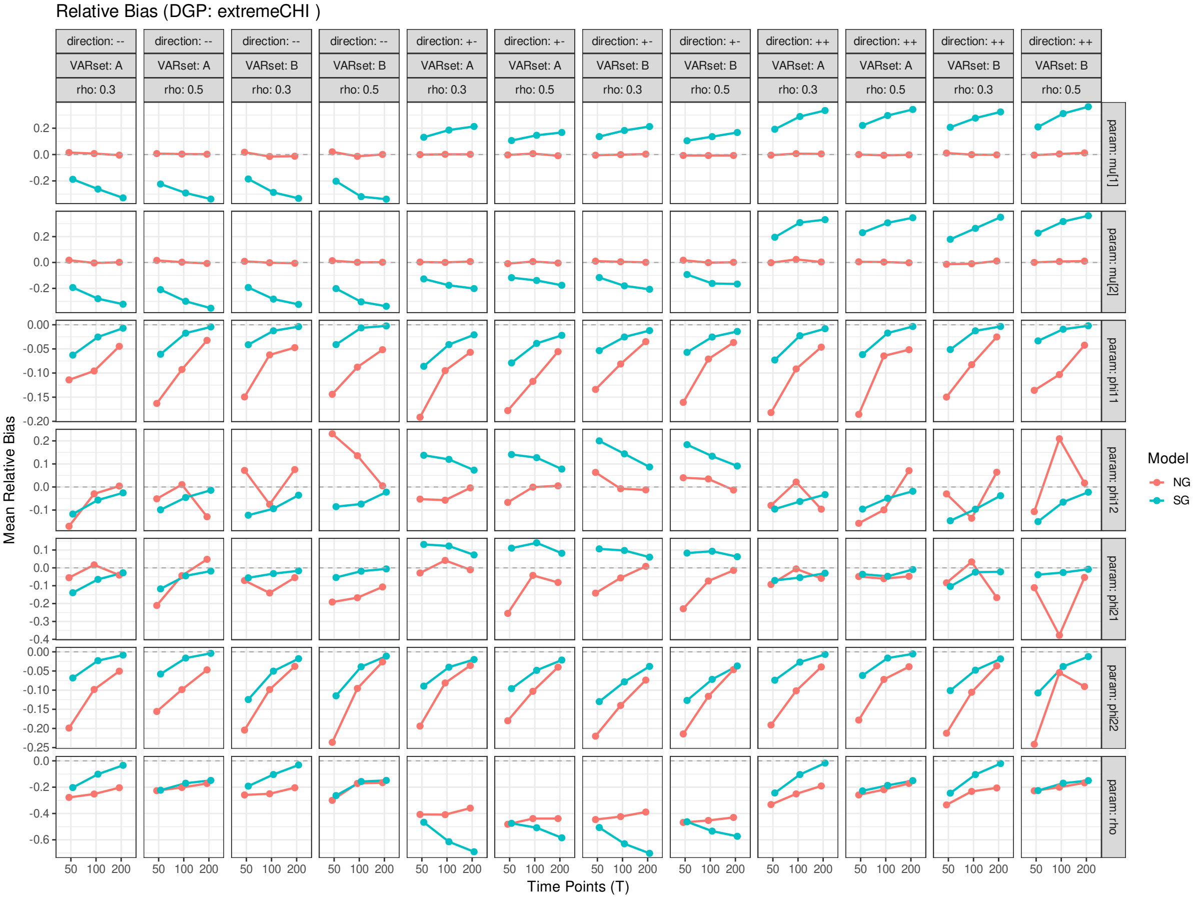
6. Condition 3: Extreme Skewness and Misspecification (extremeCHI)

DGP: standardized chi-squared ($df = 1$) innovations. Both models are marginally misspecified; SG allows skewness in the innovations but remains outside the χ^2_1 family.

```r
plots_extreme <- generate_plots_for_condition("extremeCHI")
```

6.1. Relative Bias (extremeCHI)

```r
print(plots_extreme$bias)
```



Interpretation: High Bias for SG Model in intercepts (μ_1, μ_2)

A salient feature of `extremeCHI` is the systematic bias in the SG intercepts (μ_1, μ_2), while the NG intercepts remain close to 0.

- **NG:** Although NG is severely misspecified in higher moments, the DGP innovations are standardized to have mean 0. In these simulations, NG's posterior for μ remains centered near 0, with misspecification primarily expressed through distorted estimates of Φ and, most prominently, ρ.

- **SG:** SG is also misspecified under `extremeCHI` because standardized χ^2_1 marginals lie far outside the skew-normal family (Table below). Under the centered skew-normal parameterization, the innovation distribution is constrained to have mean 0, but cannot reproduce the DGP tail behavior. Empirically, the posterior shifts μ away from 0 in some settings, suggesting that the intercept absorbs part of the marginal misspecification through joint estimation with (Φ, α, ω).

This shift in μ does not mitigate copula attenuation: ρ remains strongly biased under both models.

ℹ Note

The `extremeCHI` condition uses standardized Chi-squared innovations with 1 degree of freedom ($\chi^2(1)$). The mismatch between this DGP and the model assumptions is strong:

Distribution	Theoretical Skewness	Theoretical Excess Kurtosis
True DGP ($\chi^2(1)$)	≈ 2.83	12
NG Model (Normal)	0	0
SG Model (Skew-Normal Max)	≈ 0.995	≈ 0.869

ℹ Note

Skewness and kurtosis formulas

- χ^2_ν: $\mathbb{E} = \nu, \quad \mathrm{Var} = 2\nu, \quad \gamma_1 = \sqrt{\frac{8}{\nu}}, \quad \gamma_2 = \frac{12}{\nu}$ (excess). For $\nu = 1$: $\gamma_1 \approx 2.828$, $\gamma_2 = 12$.
- **Skew-normal** $\mathrm{SN}(\xi, \omega, \alpha)$: with $\delta = \alpha/\sqrt{1+\alpha^2}$,

$$\mu = \xi + \omega\,\delta\,\sqrt{\frac{2}{\pi}}, \quad \sigma^2 = \omega^2\Big(1 - \frac{2\delta^2}{\pi}\Big),$$

$$\gamma_1 = \frac{(4-\pi)}{2}\,\frac{\big(\delta\sqrt{\frac{2}{\pi}}\big)^3}{\big(1-\frac{2\delta^2}{\pi}\big)^{3/2}}, \qquad \gamma_2 = 2(\pi-3)\,\frac{\big(\delta\sqrt{\frac{2}{\pi}}\big)^4}{\big(1-\frac{2\delta^2}{\pi}\big)^2}.$$

As $|\alpha| \to \infty$ (i.e., $|\delta| \to 1$), the **maximum skewness** is $\gamma_1 \approx 0.995$ and the **maximum excess kurtosis** is $\gamma_2 \approx 0.869$. This is the basis for the entries in the table contrasting χ^2_1 vs. Normal vs. Skew-Normal.

Mechanism: PIT distortion under marginal misspecification. The PIT maps each margin Y to $U = F_{\text{assumed}}(Y)$. If $F_{\text{assumed}} = F_{\text{true}}$, then $U \sim \mathrm{Uniform}(0,1)$. Under misspecification: - **Tail compression:** If the true data have **heavier right tails** (e.g., χ^2_1) than the assumed SN/Normal, then very large Y values **do not land near 1** after transformation;

they are pulled back toward the center (e.g., $U \approx 0.8$ instead of 0.98). Left tails are similarly distorted under mirroring. - **Rank distortion:** The PIT is a *ranking* device. By compressing real extremes toward the middle, **co-extreme** events (Y_1, Y_2) that truly move together in the tails no longer co-locate in the **corners** of $[0,1]^2$; instead they fall into the **interior**.

For Gaussian copulas, dependence is most visible in the *corners*. When misspecified marginals push mass away from corners, the copula "sees" less tail co-movement even if it exists in the data, and any fitted ρ is forced downward. This pattern is reflected in the ρ relative-bias panels (plotted on a $[-2, 0]$ y-axis).

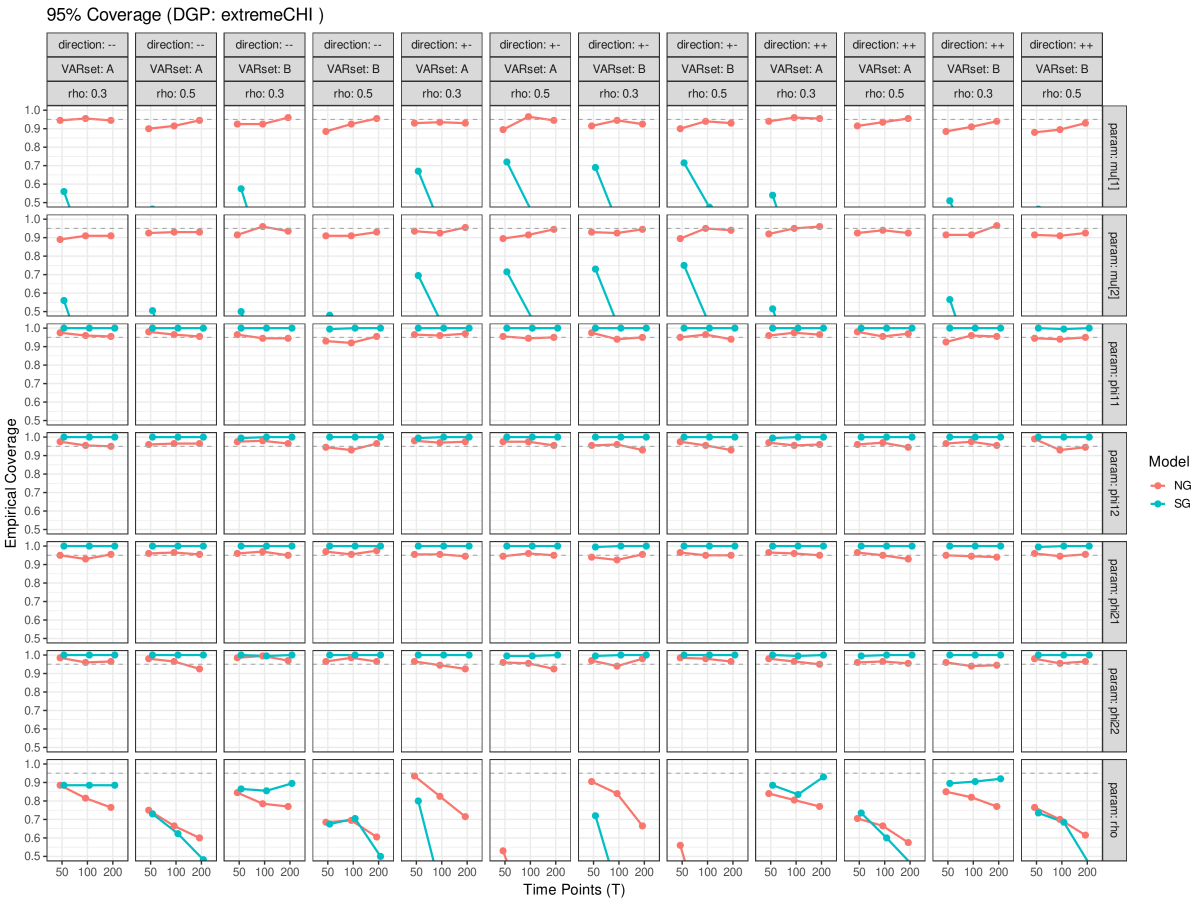
Mechanism: attenuation of ρ. The joint likelihood factorizes as

$$\prod_t c(U_{1,t}, U_{2,t}; \rho)\, f_1(y_{1,t})\, f_2(y_{2,t}),$$

so only the copula term $c(\cdot\,; \rho)$ can adjust dependence. When the PIT pushes tail pairs $(U_{1,t}, U_{2,t})$ toward the center ($\approx 0.5, 0.5$), the Gaussian copula density becomes less sensitive to ρ (the score in ρ flattens). To avoid penalizing improbable tail corners that the distorted (U_1, U_2) no longer occupy, the MLE/posterior moves ρ down toward 0. Hence the large negative relative bias and sub-nominal coverage for ρ in `extremeCHI`, regardless of "Clean" or "Problematic" MCMC status.

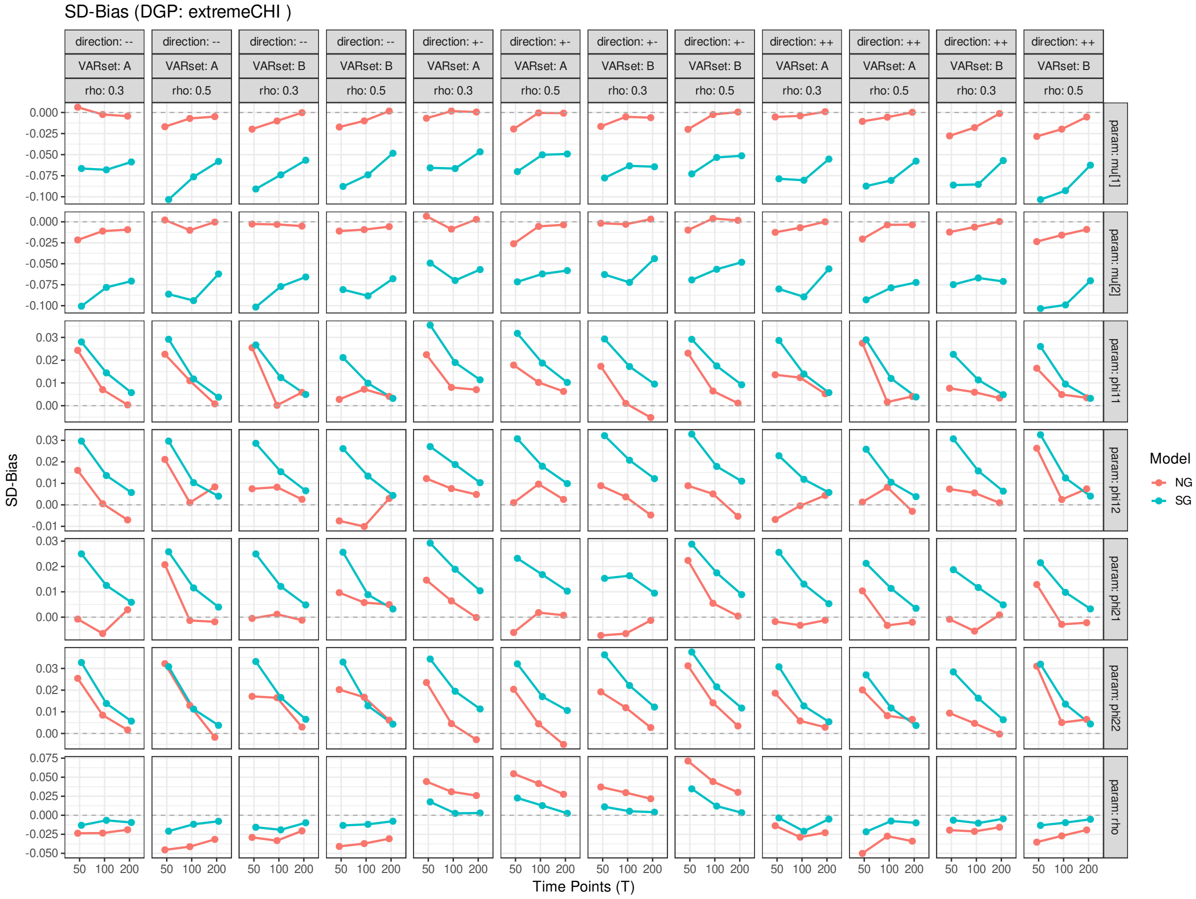
6.2. 95% Coverage (extremeCHI)

```
print(plots_extreme$coverage)
```



6.3. SD-Bias (extremeCHI)

```
print(plots_extreme$sdbias)
```



Interpretation (extremeCHI): Severe marginal misspecification produces parameter-specific failures.

- **NG:** μ remains approximately unbiased, but Φ and especially ρ exhibit substantial bias and undercoverage; in mixed-sign settings, posterior means for ρ can cross zero (relative bias < -1).
- **SG:** Φ is comparatively robust (bias smaller and coverage closer to nominal, particularly for $T \geq 100$), but μ is biased and ρ remains severely attenuated.

The shared failure for ρ is consistent with PIT distortion under extreme mismatch between the assumed marginal CDF (Normal or skew-normal) and the standardized χ^2_1 DGP. SD-bias for the dynamic parameters is often close to zero, indicating that posterior uncertainty for Φ can remain reasonably calibrated even when ρ (and, for SG, μ) is biased.

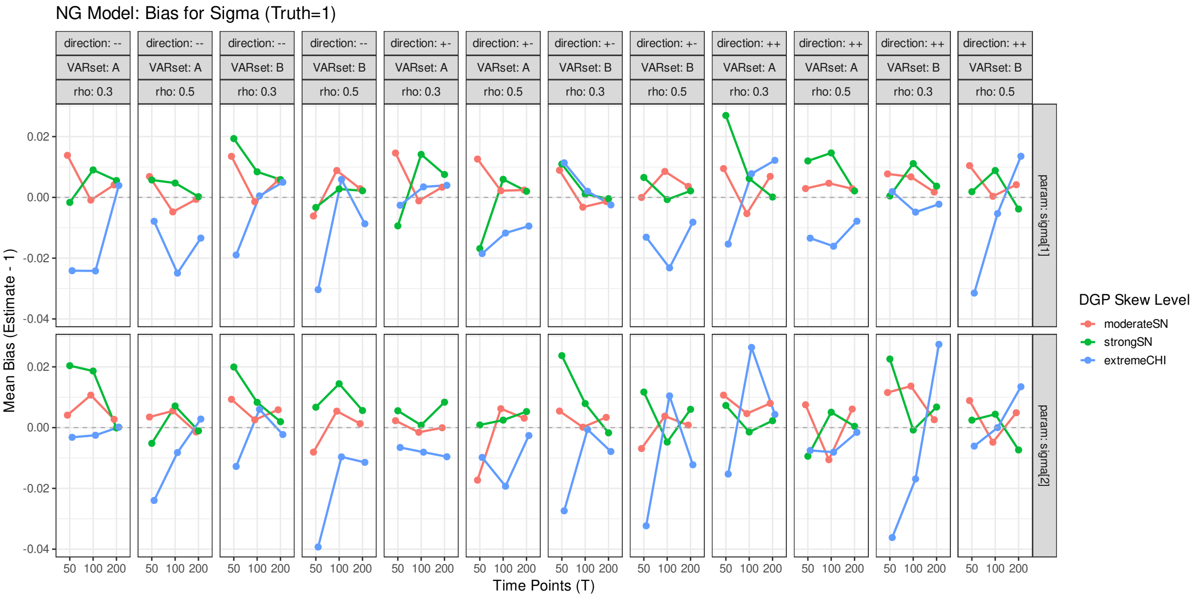
7. Cross-Condition Synthesis: Marginal Parameters

We examine the parameters governing the marginal distributions to understand the mechanisms driving the results.

7.1. NG Scale Parameter Behavior (σ)

The innovations are standardized to unit variance. Under correct specification, the NG model should recover $\sigma = 1$.

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

# We plot the absolute bias (mean_bias) here as the truth is 1.
ggplot(sigma_data, aes(x = T, y = mean_bias, color = skew_level, group = skew_level)) +
  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 +
  labs(title = "NG Model: Bias for Sigma (Truth=1)",
       y = "Mean Bias (Estimate - 1)",
       x = "Time Points (T)",
       color = "DGP Skew Level")
```



Interpretation: The σ estimates under NG show small but systematic deviations from the true value of 1 (bias $\approx \pm 0.025$). These deviations represent a byproduct of marginal misspecification: when the NG model encounters skewed or heavy-tailed innovations, it cannot capture the shape mismatch and instead makes minor adjustments to the scale parameter.

Importantly, **σ bias is not the mechanism driving ρ attenuation**. The two phenomena arise from the same root cause (marginal misspecification) but through different pathways:

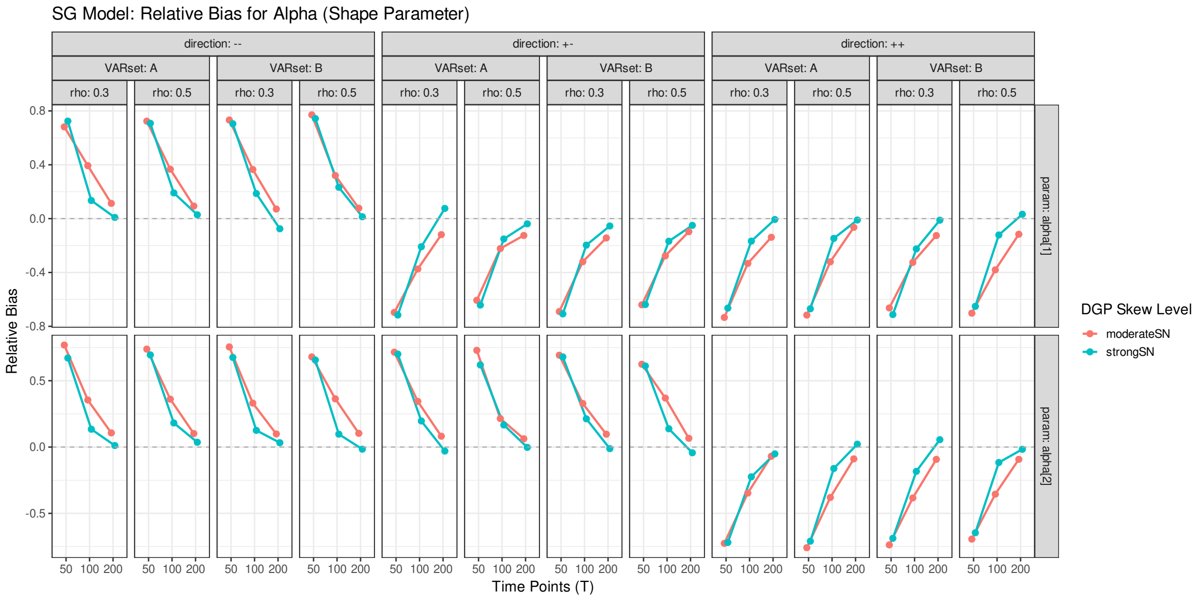
- **σ bias**: The NG model slightly adjusts scale to improve marginal likelihood under non-Gaussian data. This is a local accommodation with limited impact on other parameters.
- **ρ attenuation**: Arises from PIT distortion—regardless of how well the marginal *density* fits, if the marginal *CDF* misrepresents tail probabilities, the copula receives distorted inputs. Even with σ perfectly estimated, the PIT would still compress tail observations toward the center of $[0, 1]^2$, attenuating perceived dependence.

The small magnitude of σ bias ($\approx 2.5\%$) compared to the large ρ bias (often $> 100\%$) confirms that these are parallel consequences of misspecification rather than a causal chain where σ error propagates to ρ.

7.2. SG Shape Parameter Recovery (α)

We examine how well the SG model recovers the true shape parameter α (applicable for SN conditions).

```
alpha_data <- cond |>
  filter(param %in% c("alpha[1]", "alpha[2]"),
         Model == "SG",
         skew_level %in% c("moderateSN", "strongSN"))

# Filter NAs in case truth was unavailable
alpha_data <- alpha_data |> filter(!is.na(mean_rel_bias))

if (nrow(alpha_data) > 0) {
    ggplot(alpha_data, aes(x = T, y = mean_rel_bias, color = skew_level, group = skew_level)
      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") +
      ggh4x::facet_nested(param ~ direction + VARset + rho, labeller = label_both, scales = 
      theme_standard +
      labs(title = "SG Model: Relative Bias for Alpha (Shape Parameter)",
           y = "Relative Bias",
           x = "Time Points (T)",
              color = "DGP Skew Level")
}
```



Interpretation (shape recovery). Under the correctly specified SN DGPs, posterior means for α are biased toward 0 at $T = 50$, with substantially reduced bias at $T \geq 100$. Despite imperfect recovery of α in short series, allowing skewness in the marginals appears sufficient to stabilize estimation of the VAR dynamics relative to NG.

! Important

Caveat: large $|\alpha|$ is weakly identified at short T
For $|\alpha| \in \{4, 9\}$, the transformed shape parameter $\delta = \alpha/\sqrt{1+\alpha^2}$ is close to the boundary ± 1. In this regime, α and ω are strongly confounded and the likelihood contains limited information about the exact magnitude of α at $T = 50$. With regularizing priors, posterior estimates can therefore shrink toward 0 even under correct specification. The observed downward bias in $|\alpha|$ at $T = 50$ should be interpreted primarily as finite-sample regularization, not as a failure to recover the direction of skewness.

Mechanistic interpretation. Even when $|\alpha|$ is underestimated at $T = 50$, the sign is typically correct, which partially aligns the PIT with the data. This partial correction is consistent with the small bias and near-nominal coverage for Φ under SG in the SN DGPs, and with the improvement in Φ performance as T increases and α becomes better identified.

8. Impact of MCMC Diagnostics

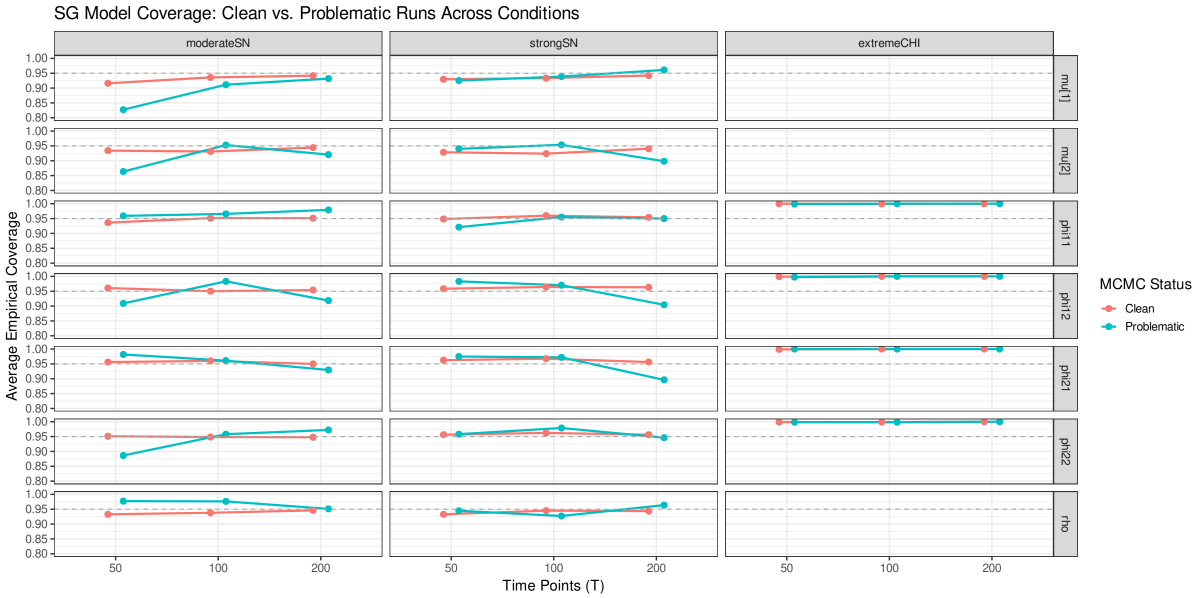
The SG model frequently encountered "Problematic" MCMC runs. We investigate if the statistical performance differs between "Clean" and "Problematic" runs for the SG model.

```r
# Helper function to re-aggregate metrics, filtering by MCMC status
# It is crucial to recalculate Empirical SD, SD-Bias, and RMSE within the subgroups.
aggregate_by_status <- function(df) {
  df |>
    # Filter out total failures
    filter(mcmc_status != "Failed/Error") |>
    group_by(condition_id, Model, param, mcmc_status, T, skew_level, direction, VARset, rho,
    summarise(
      N_valid = n(),
      mean_rel_bias = mean(rel_bias, na.rm = TRUE),
      coverage_95 = mean(cover95, na.rm = TRUE),
      # Recalculate components for SD-Bias and RMSE within the status group
      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(
        # Handle cases where N_valid=1, leading to NA emp_sd
        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)
```

8.1. Coverage Split by MCMC Status (SG Model)

We visualize the coverage across all three conditions, comparing Clean vs. Problematic runs.

```r
status_comparison_data <- cond_status |>
  filter(Model == "SG",
         param %in% core_params)

# We focus the visualization on the interaction between T, Status, and Skew Level
# We average over VARset, rho, and direction for a clearer overview
status_overview <- status_comparison_data |>
  group_by(T, param, skew_level, mcmc_status) |>
  summarise(mean_coverage = mean(coverage_95, na.rm = TRUE), .groups = 'drop')


ggplot(status_overview,
       aes(x = T, y = mean_coverage, 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 by parameter and skew level
  facet_grid(param ~ skew_level) +
  theme_standard +
  labs(title = "SG Model Coverage: Clean vs. Problematic Runs Across Conditions",
       y = "Average Empirical Coverage",
       x = "Time Points (T)",
       color = "MCMC Status") +
  coord_cartesian(ylim = c(0.8, 1.0))
```



Interpretation: Impact of MCMC Status Across this design, SG coverage for Φ is similar in the *Clean* and *Problematic* subsets; even under `extremeCHI`, coverage for the dynamic parameters remains near nominal, whereas coverage for μ deteriorates (Section 6). This indicates that—at least for the functionals assessed here—the diagnostic rule used to define *Problematic* status is not, on average, strongly associated with degraded Φ coverage.

Caution

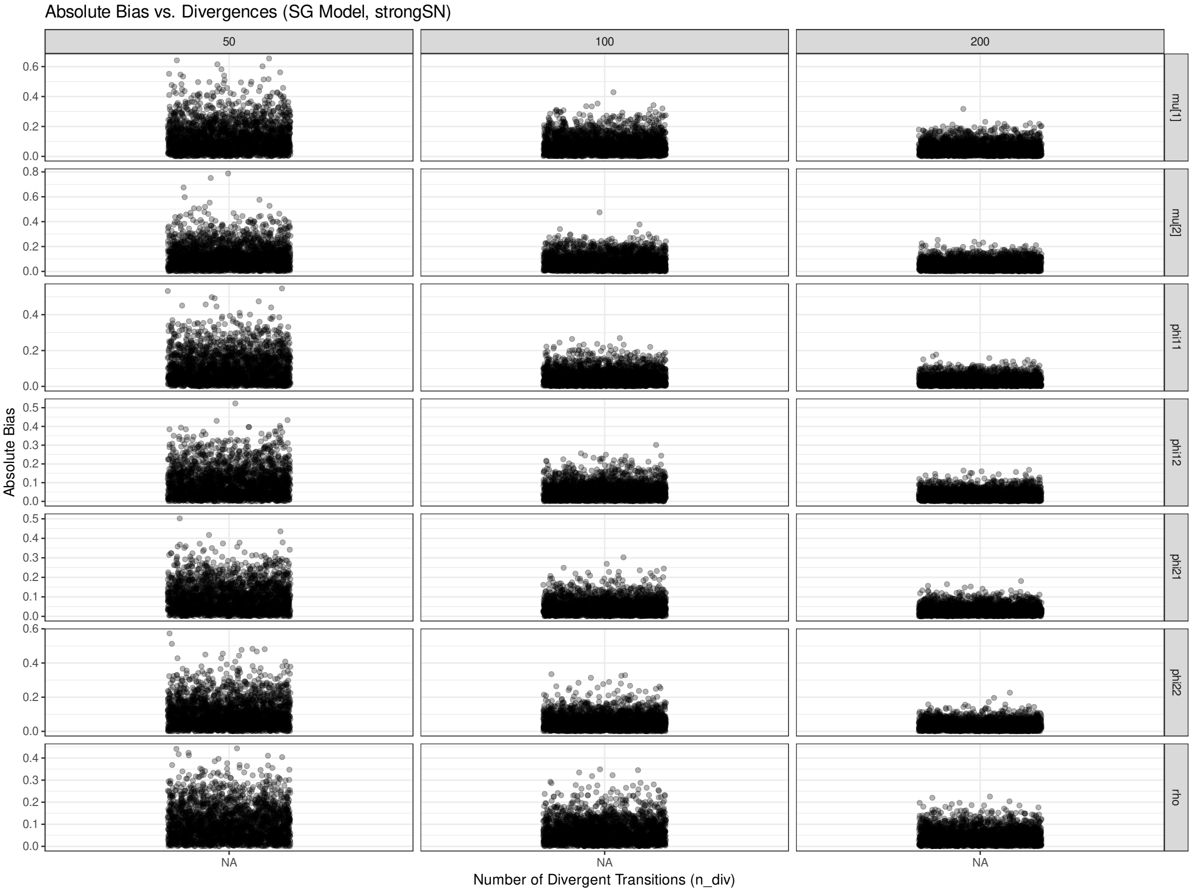
Caveats on interpreting “Clean” vs. “Problematic” comparisons

1. *Problematic* is a composite label ($\widehat{R} > 1.01$ and/or $n_{\text{div}} > 0$). Pooling these sources of pathology can dilute relationships between diagnostics and inferential quality.
2. Coverage is computed conditional on runs that completed sampling; it does not account for selection induced by failed fits.
3. The summaries average over ρ, VAR set, and skewness direction. Localized failures in specific design cells may be obscured by aggregation.
4. Divergences can bias some posterior functionals even when marginal coverage for a subset of parameters appears adequate; they should continue to be treated as a warning sign requiring model reparameterization or stricter sampling settings.

8.2. Relationship between Bias and Divergences

We examine if runs with more divergences exhibit higher bias at the replication level. We focus on the SG model under `strongSN` conditions.

```r
div_bias_data <- rep_df |>
  filter(Model == "SG",
         skew_level == "strongSN",
         param %in% core_params,
         mcmc_status != "Failed/Error")

# Use absolute bias for comparison
ggplot(div_bias_data, aes(x = n_div, y = abs(bias))) +
  geom_point(alpha = 0.3, position = position_jitter(width = 0.2), size = 2) +
  # Use a generalized additive model (gam) for smoothing to capture non-linear relationships
  geom_smooth(method = "gam", color = "red", linewidth = 1.5) +
  facet_grid(param ~ T, scales = "free") +
  theme_standard +
  labs(title = "Absolute Bias vs. Divergences (SG Model, strongSN)",
       x = "Number of Divergent Transitions (n_div)",
       y = "Absolute Bias")
```



Interpretation: Bias vs. Divergences There does not appear to be a strong correlation between the number of divergences and the absolute bias for the core parameters. High bias occurs in runs with few divergences, and low bias occurs in runs with many divergences. This reinforces the conclusion that MCMC diagnostics, while important indicators of computational issues, are not reliable predictors of statistical accuracy in this context.

9. Exporting Key Tables

```
# 1. Main Condition-Level Summary (Aggregated across all successful runs)
# This table includes all metrics (Bias, Coverage, SD-Bias, RMSE, Posterior SD, Divergences)
# aggregated by condition, model, and parameter.

export_cond <- cond |>
  # Select relevant columns and restore original factor values for clarity
  select(condition_id, Model, param, 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.csv"))
# message("Exported aggregated summary to: ", file.path(EXPORT_DIR, "analysis_summary_aggrega


# 2. Status-Split Summary (Aggregated within Clean/Problematic groups)
# This table allows for analyzing the impact of MCMC diagnostics on performance.

export_status <- cond_status |>
    # Select relevant columns
  select(condition_id, Model, param, mcmc_status,
         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.csv"))
# message("Exported status-split summary to: ", file.path(EXPORT_DIR, "analysis_summary_statu

# 3. MCMC Health Summary (Counts)
mcmc_health_export <- mcmc_summary |>
  pivot_wider(names_from = mcmc_status, values_from = Count, values_fill = list(Count = 0))
  arrange(Model, skew_level, T)

write_csv(mcmc_health_export, file.path(EXPORT_DIR, "analysis_mcmc_health_counts.csv"))
# message("Exported MCMC health counts to: ", file.path(EXPORT_DIR, "analysis_mcmc_health_cou
```