

PymcRltReport

Fiona Zhu

2025-11-13

```
library(rlist)
library(ez)
library(tidyverse)

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## vforcats   1.0.1     v stringr   1.5.2
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyrr    1.3.1
## v purrr    1.1.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(latex2exp)
library(lsr)
library(rticles)
library(DescTools)
library(heplots)

## Loading required package: broom
library(lme4)

## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyrr':
## 
##     expand, pack, unpack

library(rstatix)

##
## Attaching package: 'rstatix'
##
## The following object is masked from 'package:stats':
## 
##     filter

library(boot)
library(ggpubr)
library(reshape2)
```

```

## 
## Attaching package: 'reshape2'
## 
## The following object is masked from 'package:tidyverse':
## 
##     smiths

library(ggplot2)
library(readr)
source('mytheme.R')

```

Setting

```

modelname = "FreeParameters" #"Experimentwise" #"FreeParameters"
filepath = paste0("../data/", modelname, "/")
#
order_exp <- c("Encoding", "Reproduction", "Baseline", "Both", "Both_gap")
exp_labels <- c(
  "Encoding"      = "Exp. 1\nEncoding",
  "Reproduction" = "Exp. 2\nReproduction",
  "Baseline"      = "Exp. 3\nBaseline",
  "Both"          = "Exp. 4\nBoth",
  "Both_gap"      = "Exp. 5\nBoth (with gap)"
)
Exp.labs.2lines <- c("Exp. 1\nEncoding", "Exp. 2\nReproduction", "Exp. 3\nBaseline", "Exp. 4\nBoth", "E"

```

Load data

```

mdat_all <- read_csv(paste0(filepath, modelname, "_mdat_all.csv"))

## New names:
## Rows: 1440 Columns: 15
## -- Column specification
## -----
## (2): WMSize, Exp dbl (13): ...1, Unnamed: 0, NSub, curDur, repDur_mean,
## repDur_std, mPred, sd...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `--> `...1`

mdat_all$Exp <- factor(mdat_all$Exp, levels = order_exp)
mdat_all$Gap <- factor(mdat_all$Gap, labels = c("short", "long"))
mdat_all$WMSize <- factor(mdat_all$WMSize, labels = c("low", "medium", "high"))

# Here the predicted RP was based on function predict_single_subject
PredRPLList <- read_csv(paste0(filepath, modelname, "_PredRPLList.csv"))

```

```

## New names:
## Rows: 67680 Columns: 7
## -- Column specification
## -----
## (1): Exp dbl (6): ...1, curDur, Gap, WMSize, mu_r, NSub
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `--> `...1`

```

```

PredRPLList$WMSize <- factor(PredRPLList$WMSize, labels = c("low", "medium", "high"))
PredRPLList$Gap <- factor(PredRPLList$Gap, labels = c("short", "long"))

PredRPLList$Exp <- factor(PredRPLList$Exp, levels = order_exp)

```

Estimated parameters

```

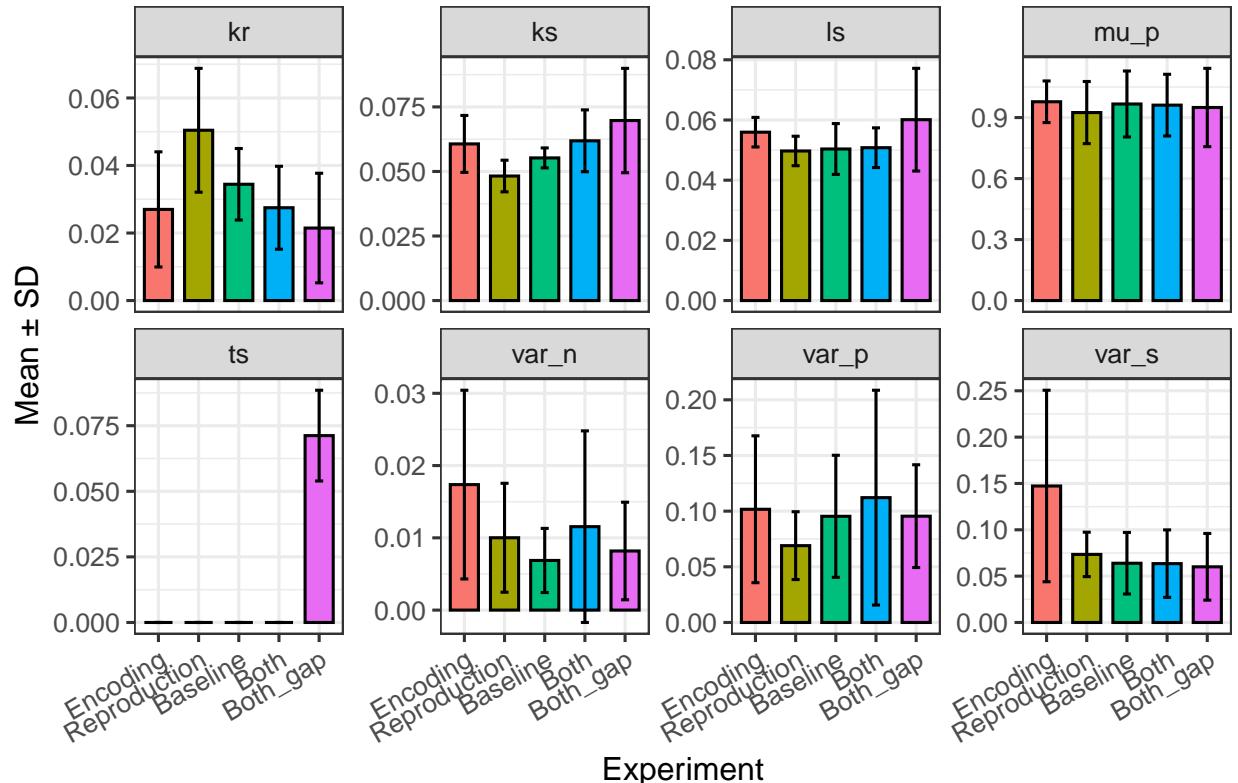
#
ParaList <- read_csv(paste0(filepath, modelname, "_ParaList.csv"))

## New names:
## Rows: 288 Columns: 17
## -- Column specification
## ----- Delimiter: ","
## (1): Exp dbl (16): ...1, ks, ls, kr, ts, var_s, mu_log, sigma_log, var_n, mu_p,
## var_p...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `--> `...1`

ParaList$Gap <- factor(ParaList$Gap, labels = c("short", "long"))
ParaList$Exp <- factor(ParaList$Exp, levels = order_exp)
ParaList$WMSize <- factor(ParaList$WMSize, labels = c("low", "medium", "high"))

```

Mean \pm SD of Parameters by Experiment



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```

# --- param_labels named list of expressions
param_labels <- list(
  ks    = expression(k[s]),
  ls    = expression(l[s]),
  kr    = expression(k[r]),
  ts    = expression(t[s]),
  var_s = expression(var[s]),
  mu_p  = expression(mu[p]),
  var_p = expression(var[p]),
  var_n = expression(var[n])
)

#
params <- c("ks", "ls", "kr", "ts", "var_s", "mu_p", "var_p", "var_n")

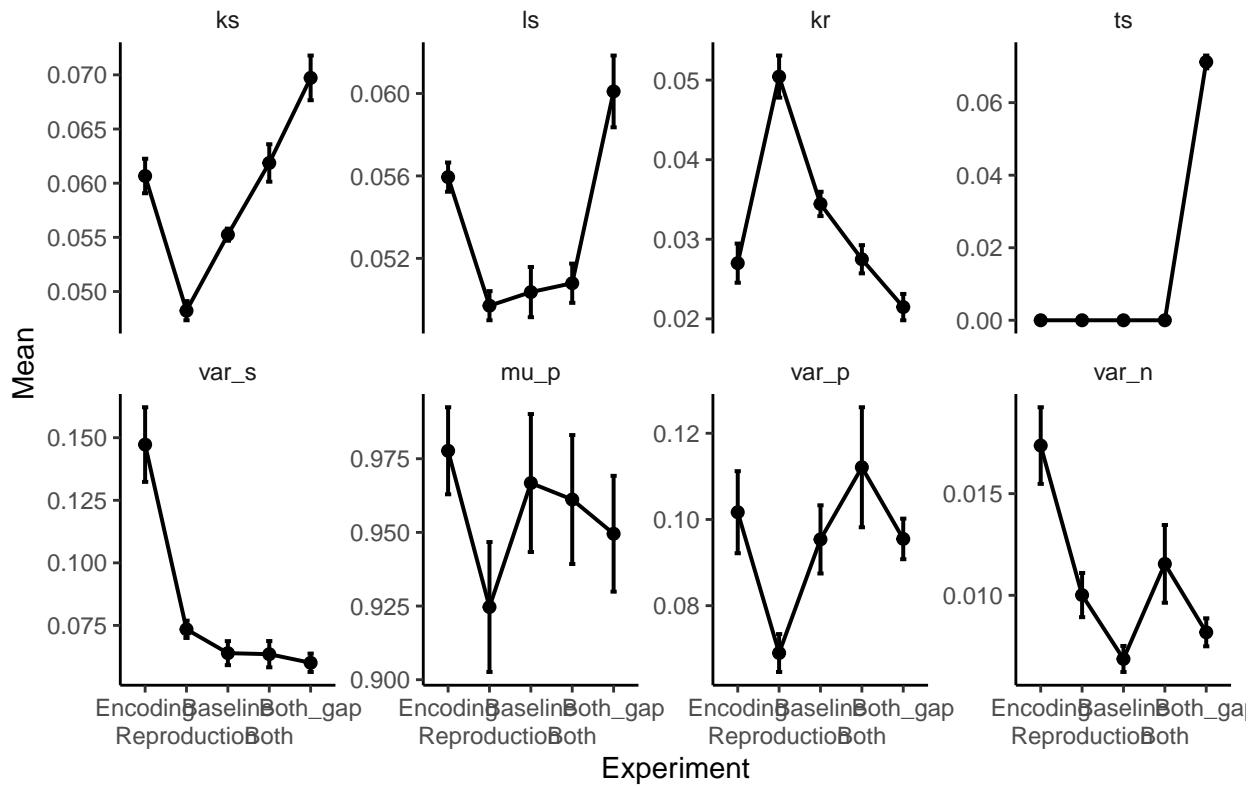
#
      SE
df_summary <- ParaList %>%
  pivot_longer(cols = all_of(params), names_to = "Parameter", values_to = "Value") %>%
  group_by(Exp, Parameter) %>%
  summarise(
    Mean = mean(Value, na.rm = TRUE),
    n = sum(!is.na(Value)),
    SD = sd(Value, na.rm = TRUE),
    SE = SD / sqrt(n),
    .groups = "drop"
  )

# Parameter      param_labels
df_summary$Parameter <- factor(df_summary$Parameter, levels = names(param_labels))

#
ggplot(df_summary, aes(x = Exp, y = Mean, group = 1)) +
  geom_line(color = "black", linewidth = 0.7) +
  geom_point(size = 1.8, color = "black") +
  geom_errorbar(aes(ymin = Mean - SE, ymax = Mean + SE),
                width = 0.15, linewidth = 0.7) +
  facet_wrap(~ Parameter, scales = "free_y", ncol = 4,
             labeller = labeller(Parameter = as_labeller(param_labels))) +
  scale_x_discrete(guide = guide_axis(n.dodge = 2)) +  # x
  theme_bw(base_size = 12) +
  theme(
    axis.text.x = element_text(angle = 30, hjust = 1),
    legend.position = "none",
    strip.text = element_text(size = 11)
  ) +
  labs(
    title = "Mean ± SE of Parameters by Experiment",
    x = "Experiment",
    y = "Mean"
  ) +
  theme_new

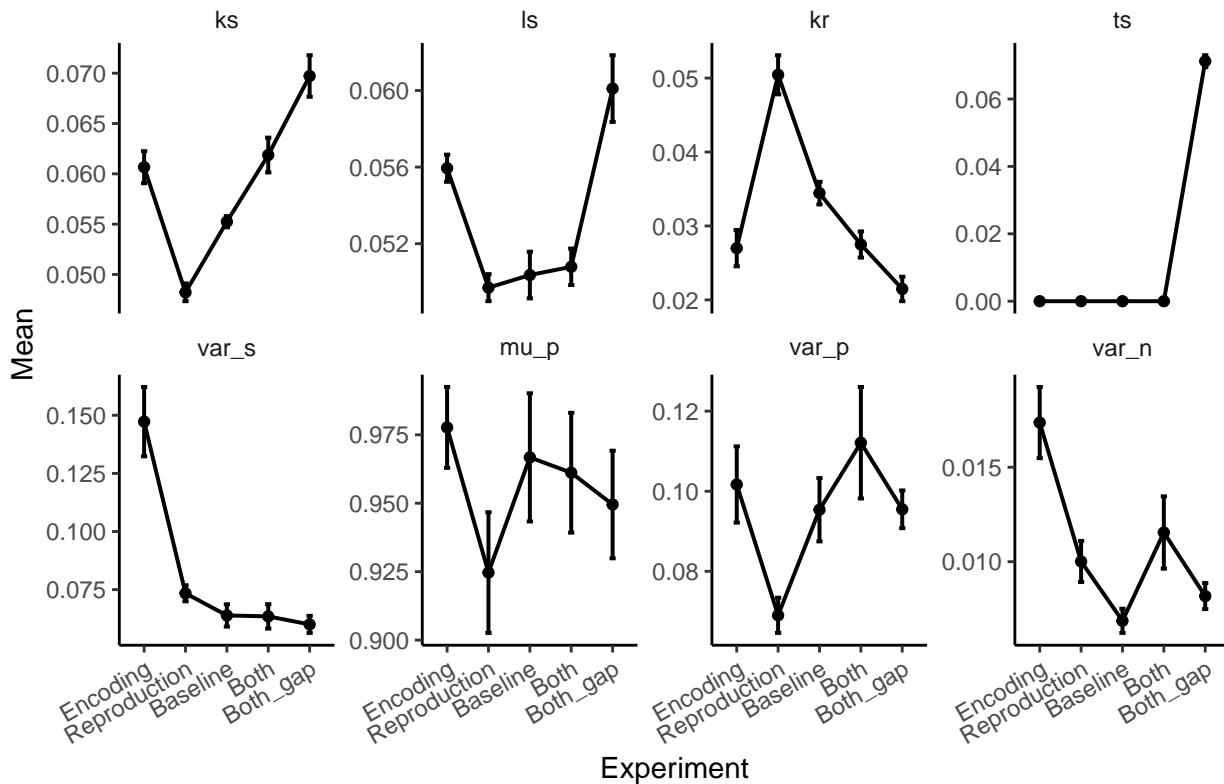
```

Mean ± SE of Parameters by Experiment



```
#> ggplot(df_summary, aes(x = Exp, y = Mean, group = Parameter)) +
#>   geom_line(color = "black", linewidth = 0.7) +
#>   geom_point(size = 1.5, color = "black") +
#>   geom_errorbar(aes(ymax = Mean + SE, ymin = Mean - SE),
#>                 width = 0.15, linewidth = 0.7) +
#>   facet_wrap(~ Parameter, scales = "free_y", ncol = 4,
#>              labeller = labeller(Parameter = as_labeller(param_labels, label_parsed))) +
#>   theme_bw(base_size = 12) +
#>   theme(
#>     axis.text.x = element_text(angle = 30, hjust = 1),
#>     legend.position = "none",
#>     strip.text = element_text(size = 11)
#>   ) +
#>   labs(
#>     title = "Mean ± SE of Parameters by Experiment",
#>     x = "Experiment",
#>     y = "Mean"
#>   ) +
#>   theme_new+scale_x_discrete(guide = guide_axis(n.dodge = 1)) +
#>   theme(axis.text.x = element_text(angle = 30, hjust = 1))
```

Mean ± SE of Parameters by Experiment



```

## predicted bias
# Here the predicted RP was based on function predict_newdata_from_posterior

newdat_all <- read_csv(paste0(filepath, modelname, "_newdat_all.csv")) %>%
  filter(!(Exp != "Both_gap" & Gap == 2.5))

## New names:
## Rows: 67680 Columns: 12
## -- Column specification
## -----
##   Delimiter: ","
##   (1): Exp dbl (11): ...1, NSub, xnew, WMSize, Gap, mPred, sdPred, CI_low,
##   CI_high, pre...
##   i Use `spec()` to retrieve the full column specification for this data. i
##   Specify the column types or set `show_col_types = FALSE` to quiet this message.
##   * `-->` `...1`

newdat_all$cv = newdat_all$sdPred/newdat_all$mPred
newdat_all$Exp <- factor(newdat_all$Exp, levels = order_exp)
newdat_all$Gap <- factor(newdat_all$Gap, labels = c("short", "long"))
newdat_all$WMSize <- factor(newdat_all$WMSize, labels = c("low", "medium", "high"))
newdat_all <- newdat_all %>%
  rename(curDur = xnew)

RP_bias_bw <- ggplot(data = mdat_all %>%
  dplyr::group_by(Exp, curDur, WMSize, Gap) %>%
  dplyr::summarize(m_repDur_mean = mean(repDur_mean), n= n(), se_repDur_mean =
  aes(x = curDur, y = m_repDur_mean - curDur,
  group = interaction(Gap, WMSize),

```

```

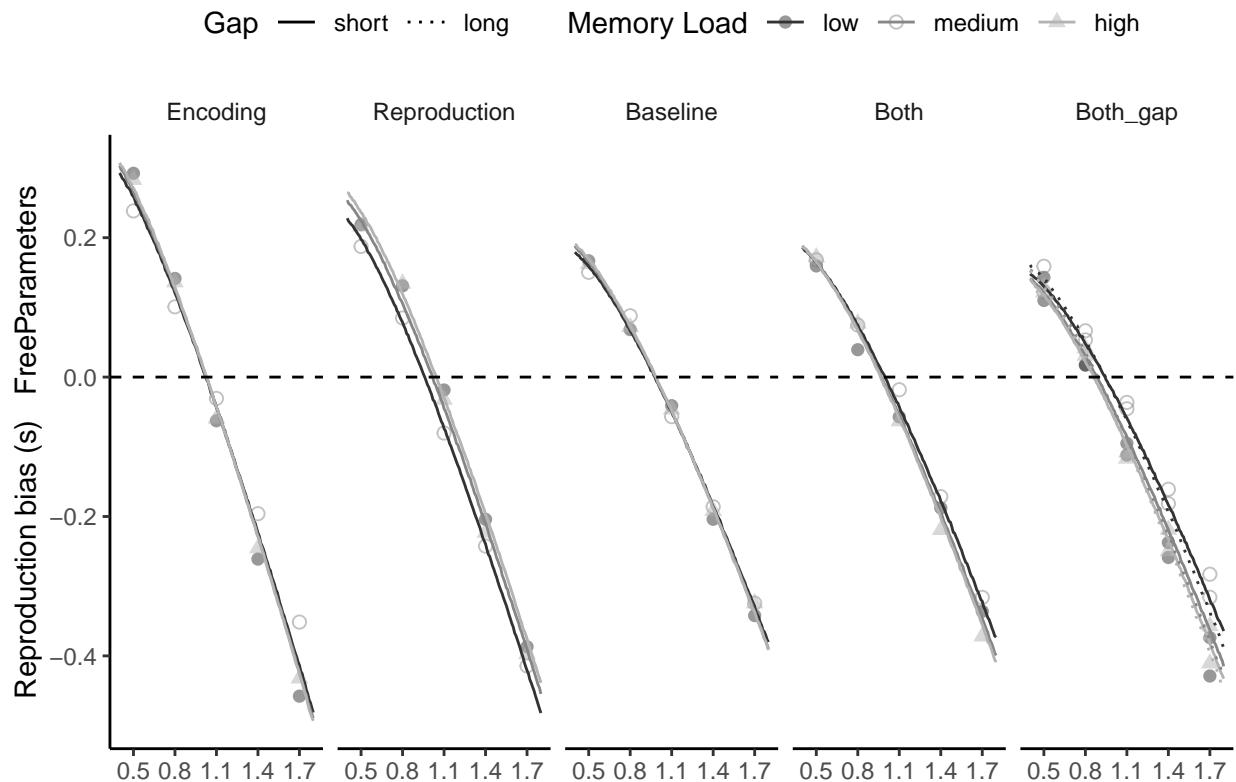
            color = as.factor(WMSize),
            shape = as.factor(WMSize))) +
geom_point(size = 2, alpha = 0.5) +
geom_line(data = newdat_all %>% dplyr::group_by(Exp, curDur, WMSize, Gap) %>% dplyr::summarize(mmPred
geom_hline(yintercept = 0, linetype = "dashed") + scale_x_continuous(breaks=c(0.5, 0.8, 1.1, 1.4, 1.7))

facet_grid(cols = vars(Exp)) +
scale_color_grey(start = 0.2, end = 0.7) +
scale_shape_manual(values = c(16, 1, 17, 2, 15, 0)) +
scale_linetype_manual(values = c("solid", "dotted")) +
labs(x = " ",
y = paste0("Reproduction bias (s)    ", modelname),
shape = "Memory Load",
linetype = "Gap",
color = "Memory Load") +
theme_new +
theme(legend.position = "top")

## `summarise()` has grouped output by 'Exp', 'curDur', 'WMSize'. You can override
## using the `.groups` argument.
## `summarise()` has grouped output by 'Exp', 'curDur', 'WMSize'. You can override
## using the `.groups` argument.

RP_bias_bw

```



```
ggsave(paste0(getwd(), "/figures/RP_bias_bw_", modelname, ".png"), RP_bias_bw, width = 8, height = 4)
```

predicted CV

```
cv_newY <- newdat_all %>%
  group_by(Exp, curDur, WMSize, Gap) %>%
  summarize(m_cv = mean(cv)) %>%
  arrange(Exp, WMSize, Gap, curDur)

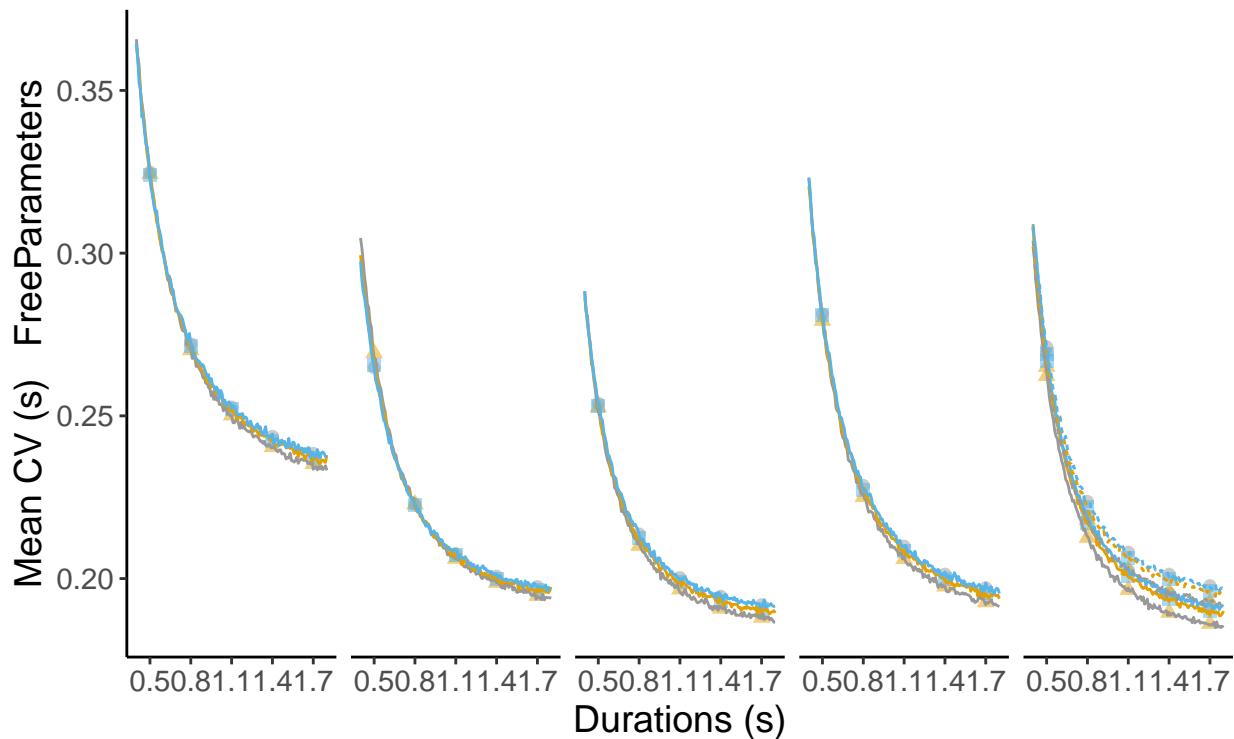
## `summarise()` has grouped output by 'Exp', 'curDur', 'WMSize'. You can override
## using the `.groups` argument.

RP_CV <- ggplot(data= mdat_all%>%
  dplyr::group_by(Exp, curDur, WMSize, Gap) %>%
  dplyr::summarize(m_predCV = mean(predCV)),
  aes(x=curDur, y= m_predCV, group = interaction(Gap, WMSize),
      color = as.factor(WMSize), shape = as.factor(WMSize))) +
  geom_point(size=2, alpha = 0.5) +
  geom_line(data = cv_newY, aes(x=curDur, y=m_cv, group = interaction(WMSize, Gap),
      linetype = Gap, color=WMSize)) +
  scale_x_continuous(breaks=c(0.5, 0.8, 1.1, 1.4, 1.7)) +
  facet_grid(~Exp) +
  labs(x="Durations (s)", y=paste0("Mean CV (s)", " ", modelname), shape="Memory Load", linetype = "G
  color = "Memory Load") + theme_new + colorSet3 +
  theme(legend.position = "top",
  legend.title = element_text(size = 14, face = "bold"),
  legend.text = element_text(size = 14),
  axis.title = element_text(size = 14),
  axis.text = element_text(size = 11),
  strip.text.x = element_blank(),
  plot.title = element_text(hjust = 0.5))

## `summarise()` has grouped output by 'Exp', 'curDur', 'WMSize'. You can override
## using the `.groups` argument.

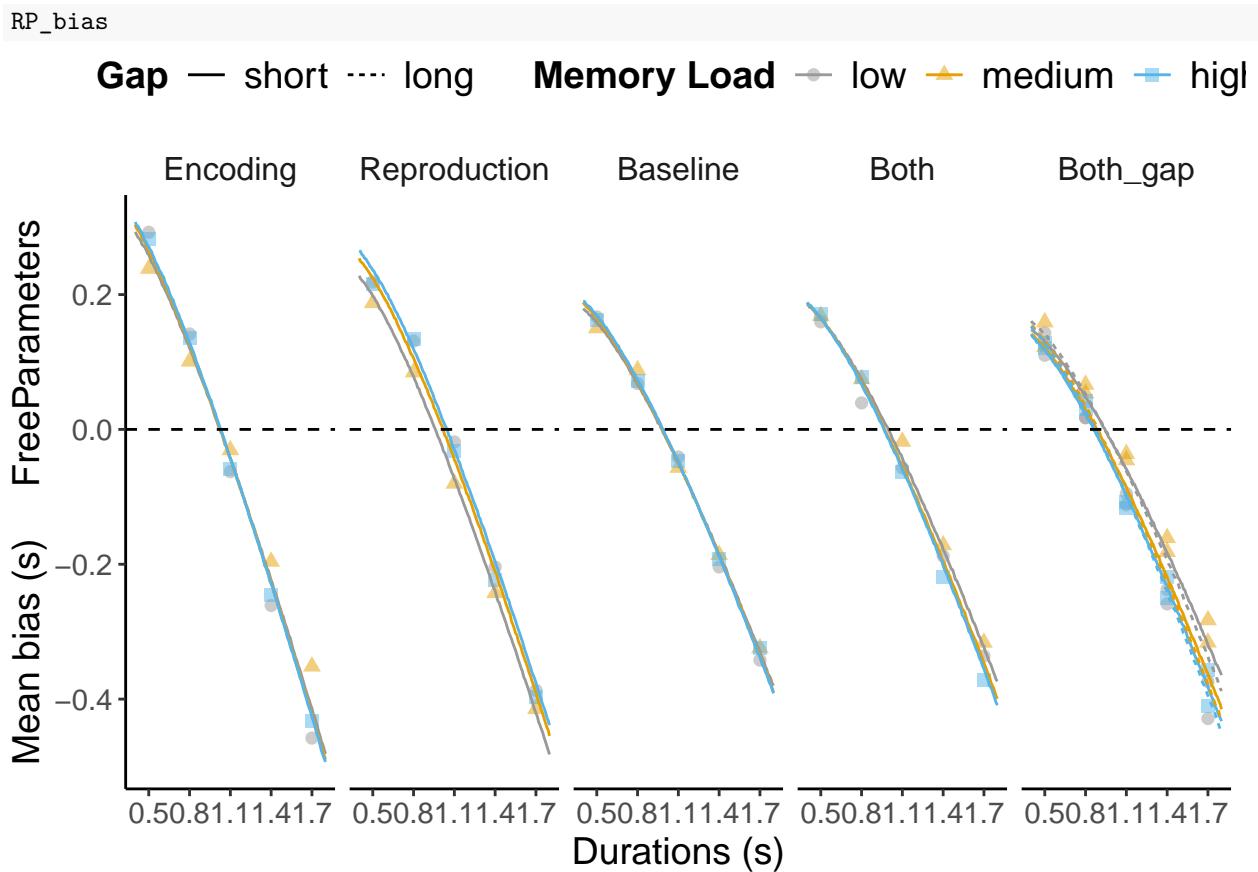
RP_CV
```

Gap — short ---- long **Memory Load** — low ▲ medium ■ high



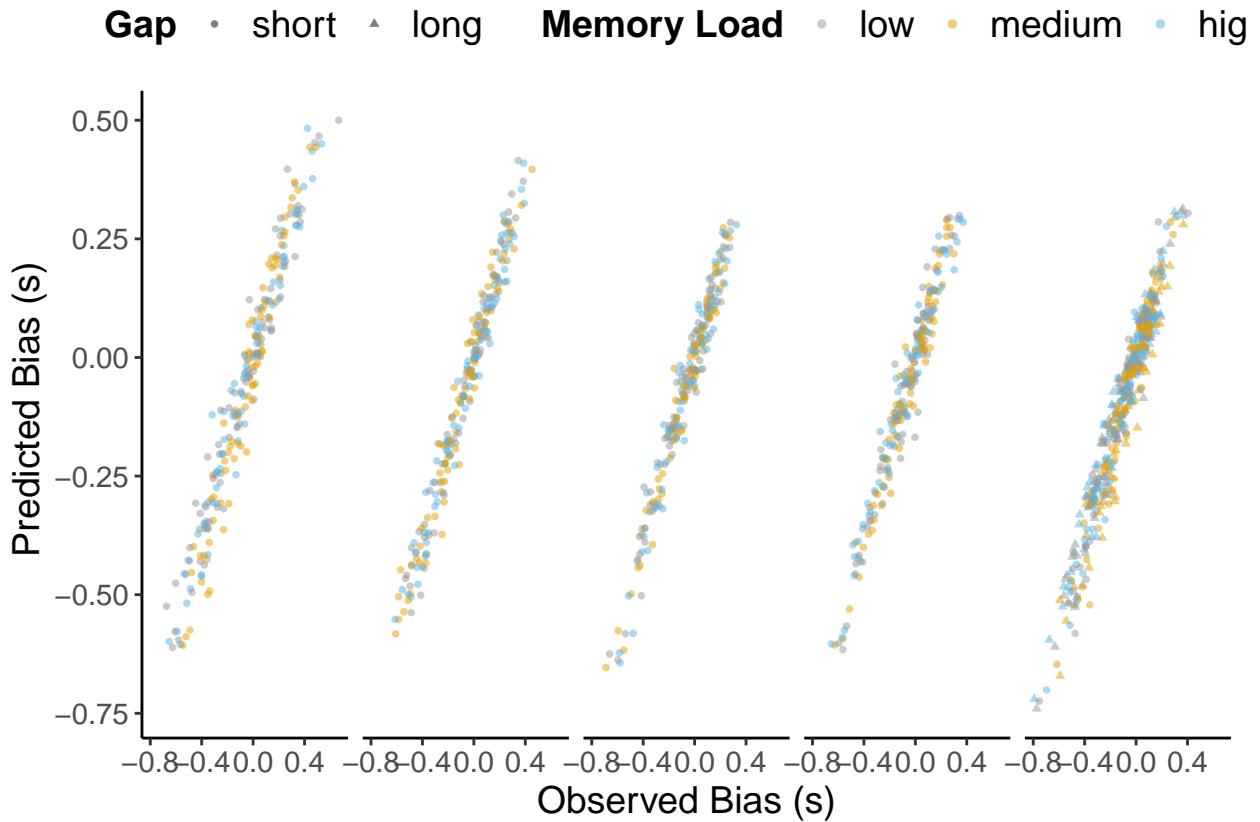
```
# plot color plots for Appendix
RP_bias <- ggplot(data = mdat_all %>%
  dplyr::group_by(Exp, curDur, WMSize, Gap) %>% dplyr::summarize(m_repDur = mean(.,
  aes(x = curDur, y = m_repDur - curDur, group = interaction(Gap, WMSize),
  color=as.factor(WMSize), shape = as.factor(WMSize))) +
  geom_point(size=2, alpha = 0.5) +
  geom_line(data= newdat_all %>% dplyr::group_by(Exp, curDur, WMSize, Gap) %>% dplyr::summarize(mmPred =
    linetype = Gap, color=WMSize)) +
  scale_x_continuous(breaks=c(0.5, 0.8, 1.1, 1.4, 1.7)) +
  geom_hline(yintercept = 0, linetype='dashed') +
  facet_grid(cols = vars(Exp), labeller = labeller(Exp = Exp.labs.2lines)) +
  labs(x="Durations (s)", y=paste0("Mean bias (s)      ", modelname), shape ="Memory Load", linetype = "Memory Load") +
  theme_new + colorSet3 + theme(legend.position = "top",
  legend.title = element_text(size = 14, face = "bold"),
  legend.text = element_text(size = 14),
  axis.title = element_text(size = 14),
  axis.text = element_text(size = 11),
  strip.text.x = element_text(size = 12),
  plot.title = element_text(hjust = 0.5))

## `summarise()` has grouped output by 'Exp', 'curDur', 'WMSize'. You can override
## using the `groups` argument.
## `summarise()` has grouped output by 'Exp', 'curDur', 'WMSize'. You can override
## using the `groups` argument.
```

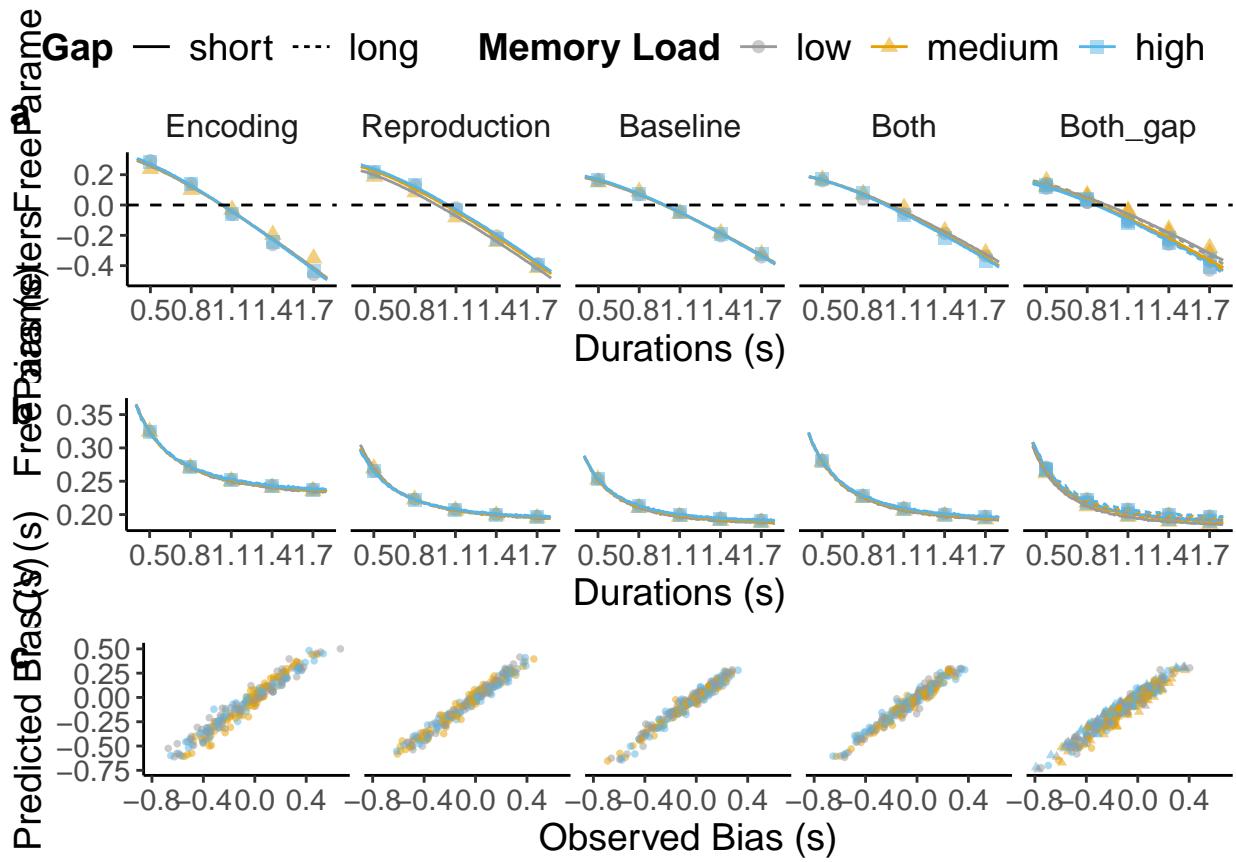


correlations of bias

```
# plot correlation coefficients r of observed and predicted bias
plt_corr_obs_pred <- ggplot(data = mdat_all,
                           aes(x = repErr, y = predErr, color = as.factor(WMSize),
                               shape = as.factor(Gap))) +
  geom_point(size=1, alpha = 0.5) +
  facet_grid(~Exp) +
  labs(x="Observed Bias (s)", y=" Predicted Bias (s)", shape="Gap",
       color = "Memory Load") + theme_new + colorSet3 +
  theme(legend.position = "top",
        legend.title = element_text(size = 14, face = "bold"),
        legend.text = element_text(size = 14),
        axis.title = element_text(size = 14),
        axis.text = element_text(size = 11),
        strip.text.x = element_blank(),
        plot.title = element_text(hjust = 0.5))
plt_corr_obs_pred
```



```
## combine predicted bias and cv
fig_bias_cv <- ggarrange(RP_bias, RP_CV, plt_corr_obs_pred, common.legend = TRUE,
                           ncol=1, nrow=3, heights = c(1.1, 0.9, 0.9),
                           labels = c("a", "b", "c"))
ggsave(paste0(getwd(), "/figures/fig_bias_cv_pred_", modelname, ".png"),
       fig_bias_cv, width = 9, height = 9)
fig_bias_cv
```



weight of prior

```
plt_wp_Gap <- ggplot(  
  data = ParaList %>%  
    dplyr::group_by(Exp, WMSize, Gap) %>%  
    dplyr::summarise(  
      m_wp = mean(w_p),  
      n = n(),  
      se_wp = sd(w_p) / sqrt(n - 1)  
    ),  
  aes(  
    x = Exp,  
    y = m_wp,  
    ymin = m_wp - se_wp,  
    ymax = m_wp + se_wp,  
    group = interaction(Exp, WMSize, Gap),  
    color = factor(WMSize),           # FIX 1  
    shape = factor(WMSize),          # FIX 2  
    linetype = factor(Gap)          # FIX 3  
  )  
) +  
  geom_line(position = position_dodge(width = 0.3)) +  
  geom_point(position = position_dodge(width = 0.3)) +  
  geom_errorbar(width = .3, position = position_dodge(width = .3)) +  
  colorSet5 +  
  labs(  
    title = "Mean Window Placement (m_wp) by Experiment (Exp), Window Size (WMSize), and Gap (Gap)",  
    subtitle = "Error bars represent standard error (se_wp). Points represent individual experiments.",  
    x_label = "Experiment (Exp)",  
    y_label = "Mean Window Placement (m_wp)",  
    x_ticks = c("A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z"),  
    y_ticks = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 248, 249, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 268, 269, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 278, 279, 279, 280, 281, 282, 283, 284, 285, 286, 287, 287, 288, 288, 289, 289, 290, 291, 292, 293, 294, 295, 296, 297, 297, 298, 298, 299, 299, 300, 301, 302, 303, 304, 305, 306, 307, 307, 308, 308, 309, 309, 310, 311, 312, 313, 314, 315, 316, 317, 317, 318, 318, 319, 319, 320, 321, 322, 323, 324, 325, 326, 327, 327, 328, 328, 329, 329, 330, 331, 332, 333, 334, 335, 336, 337, 337, 338, 338, 339, 339, 340, 341, 342, 343, 344, 345, 346, 347, 347, 348, 348, 349, 349, 350, 351, 352, 353, 354, 355, 356, 357, 357, 358, 358, 359, 359, 360, 361, 362, 363, 364, 365, 366, 367, 367, 368, 368, 369, 369, 370, 371, 372, 373, 374, 375, 376, 377, 377, 378, 378, 379, 379, 380, 381, 382, 383, 384, 385, 386, 387, 387, 388, 388, 389, 389, 390, 391, 392, 393, 394, 395, 396, 397, 397, 398, 398, 399, 399, 400, 401, 402, 403, 404, 405, 406, 407, 407, 408, 408, 409, 409, 410, 411, 412, 413, 414, 415, 416, 417, 417, 418, 418, 419, 419, 420, 421, 422, 423, 424, 425, 426, 427, 427, 428, 428, 429, 429, 430, 431, 432, 433, 434, 435, 436, 437, 437, 438, 438, 439, 439, 440, 441, 442, 443, 444, 445, 446, 447, 447, 448, 448, 449, 449, 450, 451, 452, 453, 454, 455, 456, 457, 457, 458, 458, 459, 459, 460, 461, 462, 463, 464, 465, 466, 467, 467, 468, 468, 469, 469, 470, 471, 472, 473, 474, 475, 476, 477, 477, 478, 478, 479, 479, 480, 481, 482, 483, 484, 485, 486, 487, 487, 488, 488, 489, 489, 490, 491, 492, 493, 494, 495, 496, 497, 497, 498, 498, 499, 499, 500, 501, 502, 503, 504, 505, 506, 507, 507, 508, 508, 509, 509, 510, 511, 512, 513, 514, 515, 516, 517, 517, 518, 518, 519, 519, 520, 521, 522, 523, 524, 525, 526, 527, 527, 528, 528, 529, 529, 530, 531, 532, 533, 534, 535, 536, 537, 537, 538, 538, 539, 539, 540, 541, 542, 543, 544, 545, 546, 547, 547, 548, 548, 549, 549, 550, 551, 552, 553, 554, 555, 556, 557, 557, 558, 558, 559, 559, 560, 561, 562, 563, 564, 565, 566, 567, 567, 568, 568, 569, 569, 570, 571, 572, 573, 574, 575, 576, 577, 577, 578, 578, 579, 579, 580, 581, 582, 583, 584, 585, 586, 587, 587, 588, 588, 589, 589, 590, 591, 592, 593, 594, 595, 596, 597, 597, 598, 598, 599, 599, 600, 601, 602, 603, 604, 605, 606, 607, 607, 608, 608, 609, 609, 610, 611, 612, 613, 614, 615, 616, 617, 617, 618, 618, 619, 619, 620, 621, 622, 623, 624, 625, 626, 627, 627, 628, 628, 629, 629, 630, 631, 632, 633, 634, 635, 636, 637, 637, 638, 638, 639, 639, 640, 641, 642, 643, 644, 645, 646, 647, 647, 648, 648, 649, 649, 650, 651, 652, 653, 654, 655, 656, 657, 657, 658, 658, 659, 659, 660, 661, 662, 663, 664, 665, 666, 667, 667, 668, 668, 669, 669, 670, 671, 672, 673, 674, 675, 676, 677, 677, 678, 678, 679, 679, 680, 681, 682, 683, 684, 685, 686, 687, 687, 688, 688, 689, 689, 690, 691, 692, 693, 694, 695, 696, 697, 697, 698, 698, 699, 699, 700, 701, 702, 703, 704, 705, 706, 707, 707, 708, 708, 709, 709, 710, 711, 712, 713, 714, 715, 716, 717, 717, 718, 718, 719, 719, 720, 721, 722, 723, 724, 725, 726, 727, 727, 728, 728, 729, 729, 730, 731, 732, 733, 734, 735, 736, 737, 737, 738, 738, 739, 739, 740, 741, 742, 743, 744, 745, 746, 747, 747, 748, 748, 749, 749, 750, 751, 752, 753, 754, 755, 756, 757, 757, 758, 758, 759, 759, 760, 761, 762, 763, 764, 765, 766, 767, 767, 768, 768, 769, 769, 770, 771, 772, 773, 774, 775, 776, 777, 777, 778, 778, 779, 779, 780, 781, 782, 783, 784, 785, 786, 787, 787, 788, 788, 789, 789, 790, 791, 792, 793, 794, 795, 796, 797, 797, 798, 798, 799, 799, 800, 801, 802, 803, 804, 805, 806, 807, 807, 808, 808, 809, 809, 810, 811, 812, 813, 814, 815, 816, 817, 817, 818, 818, 819, 819, 820, 821, 822, 823, 824, 825, 826, 827, 827, 828, 828, 829, 829, 830, 831, 832, 833, 834, 835, 836, 837, 837, 838, 838, 839, 839, 840, 841, 842, 843, 844, 845, 846, 847, 847, 848, 848, 849, 849, 850, 851, 852, 853, 854, 855, 856, 857, 857, 858, 858, 859, 859, 860, 861, 862, 863, 864, 865, 866, 867, 867, 868, 868, 869, 869, 870, 871, 872, 873, 874, 875, 876, 877, 877, 878, 878, 879, 879, 880, 881, 882, 883, 884, 885, 886, 887, 887, 888, 888, 889, 889, 890, 891, 892, 893, 894, 895, 896, 897, 897, 898, 898, 899, 899, 900, 901, 902, 903, 904, 905, 906, 907, 907, 908, 908, 909, 909, 910, 911, 912, 913, 914, 915, 916, 917, 917, 918, 918, 919, 919, 920, 921, 922, 923, 924, 925, 926, 927, 927, 928, 928, 929, 929, 930, 931, 932, 933, 934, 935, 936, 937, 937, 938, 938, 939, 939, 940, 941, 942, 943, 944, 945, 946, 947, 947, 948, 948, 949, 949, 950, 951, 952, 953, 954, 955, 956, 957, 957, 958, 958, 959, 959, 960, 961, 962, 963, 964, 965, 966, 967, 967, 968, 968, 969, 969, 970, 971, 972, 973, 974, 975, 976, 977, 977, 978, 978, 979, 979, 980, 981, 982, 983, 984, 985, 986, 987, 987, 988, 988, 989, 989, 990, 991, 992, 993, 994, 995, 996, 997, 997, 998, 998, 999, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1006, 1007, 1007, 1008, 1008, 1009, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1017, 1018, 1018, 1019, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1027, 1028, 1028, 1029, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1037, 1038, 1038, 1039, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1047, 1048, 1048, 1049, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1057, 1058, 1058, 1059, 1059, 1060, 1061, 1062, 1063, 1064, 1065, 1066, 1067, 1067, 1068, 1068, 1069, 1069, 1070, 1071, 1072, 1073, 1074, 1075, 1076, 1077, 1077, 1078, 1078, 1079, 1079, 1080, 1081, 1082, 1083, 1084, 1085, 1086, 1087, 1087, 1088, 1088, 1089, 1089, 1090, 1091, 1092, 1093, 1094, 1095, 1096, 1097, 1097, 1098, 1098, 1099, 1099, 1100, 1101, 1102, 1103, 1104, 1105, 1106, 1106, 1107, 1107, 1108, 1108, 1109, 1109, 1110, 1111, 1112, 1113, 1114, 1115, 1116, 1117, 1117, 1118, 1118, 1119, 1119, 1120, 1121, 1122, 1123, 1124, 1125, 1126, 1127, 1127, 1128, 1128, 1129, 1129, 1130, 1131, 1132, 1133, 1134, 1135, 1136, 1137, 1137, 1138, 1138, 1139, 1139, 1140, 1141, 1142, 1143, 1144, 1145, 1146, 1147, 1147, 1148, 1148, 1149, 1149, 1150, 1151, 1152, 1153, 1154, 1155, 1156, 1157, 1157, 1158, 1158, 1159, 1159, 1160, 1161, 1162, 1163, 1164, 1165, 1166, 1167, 1167, 1168, 1168, 1169, 1169, 1170, 1171, 1172, 1173, 1174, 1175, 1176, 1177, 1177, 1178, 1178, 1179, 1179, 1180, 1181, 1182, 1183, 1184, 1185, 1186, 1187, 1187, 1188, 1188, 1189, 1189, 1190, 1191, 1192, 1193, 1194, 1195, 1196, 1196, 1197, 1197, 1198, 1198, 1199, 1199, 1200, 1201, 1202, 1203, 1204, 1205, 1206, 1206, 1207, 1207, 1208, 1208, 1209, 1209, 1210, 1211, 1212, 1213, 1214, 1215, 1216, 1217, 1217, 1218, 1218, 1219, 1219, 1220, 1221, 1222, 1223, 1224, 1225, 1226, 1227, 1227, 1228, 1228, 1229, 1229, 1230, 1231, 1232, 1233, 1234, 1235, 1236, 1237, 1237, 1238, 1238, 1239, 1239, 1240, 1241, 1242, 1243, 1244, 1245, 1246, 1247, 1247, 1248, 1248, 1249, 1249, 1250, 1251, 1252, 1253, 1254, 1255, 1256, 1257, 1257, 1258, 1258, 1259, 1259, 1260, 1261, 1262, 1263, 1264, 1265, 1266, 1267, 1267, 1268, 1268, 1269, 1269, 1270, 1271, 1272, 1273, 1274, 1275, 1276, 1277, 1277, 1278, 1278, 1279, 1279, 1280, 1281, 1282, 1283, 1284, 1285, 1286, 1287, 1287, 1288, 1288, 1289, 1289, 1290, 1291, 1292, 1293, 1294, 1295, 1296, 1296, 1297, 1297, 1298, 1298, 1299, 1299, 1300, 1301, 1302, 1303, 1304, 1305, 1306, 1306, 1307, 1307, 1308, 1308, 1309, 1309, 1310, 1311, 1312, 1313, 1314, 1315, 1316, 1317, 1317, 1318, 1318, 1319, 1319, 1320, 1321, 1322, 1323, 1324, 1325, 1326, 1327, 1327, 1328, 1328, 1329, 1329, 1330, 1331, 1332, 1333, 1334, 1335, 1336, 1337, 1337, 1338, 1338, 1339, 1339, 1340, 1341, 1342, 1343, 1344, 1345, 1346, 1347, 1347, 1348, 1348, 1349, 1349, 1350, 1351, 1352, 1353, 1354, 1355, 1356, 1357, 1357, 1358, 1358, 1359, 1359, 1360, 1361, 1362, 1363, 1364, 1365, 1366, 1367, 1367, 1368, 1368, 1369, 1369, 1370, 1371, 1372, 1373, 1374, 1375, 1376, 1377, 1377, 1378, 1378, 1379, 1379, 1380, 1381, 1382, 1383, 1384, 1385, 1386, 1387, 1387, 1388, 1388, 1389, 1389, 1390, 1391, 1392, 1393, 1394, 1395, 1396, 1396, 1397, 1397, 1398, 1398, 1399, 1399, 1400, 1401, 1402, 1403, 1404, 1405, 1406, 1406, 1407, 1407, 1408, 1408, 1409, 1409, 1410, 1411, 1412, 1413, 1414, 1415, 1416, 1417, 1417, 1418, 1418, 1419, 1419, 1420, 1421, 1422, 1423, 1424, 1425, 1426, 1427, 1427, 1428, 1428, 1429, 1429, 1430, 1431, 1432, 1433, 1434, 1435, 1436, 1437, 1437, 1438, 1438, 1439, 1439, 1440, 1441, 1442, 1443, 1444, 1445, 1446, 1447, 1447, 1448, 1448, 1449, 1449, 1450, 1451, 1452, 1453, 1454, 1455, 1456, 1457, 1457, 1458, 1458, 1459, 1459, 1460, 1461, 1462, 1463, 1464, 1465, 1466, 1467, 1467, 1468, 1468, 1469, 1469, 1470, 1471, 1472, 1473, 1474, 1475, 1476, 1477, 1477, 1478, 1478, 1479, 1479, 1480, 1481, 1482, 1483, 1484, 1485, 1486, 1487, 1487, 1488, 1488, 1489, 1489, 1490, 1491, 1492, 1493, 1494, 1495, 1496, 1496, 1497, 1497, 1498, 1498, 1499, 1499, 1500, 1501, 1502, 1503, 1504, 1505, 1506, 1506, 1507, 1507, 1508, 1508, 1509, 1509, 1510, 1511, 1512, 1513, 1514, 1515, 1516, 1517, 1517, 1518, 1518, 1519, 1519, 1520, 1521, 1522, 1523, 1524, 1525, 1526, 1527, 1527, 1528, 1528, 1529, 1529, 1530, 1531, 1532, 1533, 1534, 1535, 1536, 1537, 1537, 1538, 1538, 1539, 1539, 1540, 1541, 1542, 1543, 1544, 1545, 1546, 1547, 1547, 1548, 1548, 1549, 1549, 1550, 1551, 1552, 1553, 1554, 1555, 1556, 1557, 1557, 1558, 1558, 1559, 1559, 1560, 1561, 1562, 1563, 1564, 1565, 1566, 1567, 1567, 1568, 1568, 1569, 1569, 1570, 1571, 1572, 1573, 1574, 1575, 1576, 1577, 1577, 1578, 1578, 1579, 1579, 1580, 1581, 1582, 1583, 1584, 1585, 1586, 1587, 1587, 1588, 1588, 1589, 1589, 1590, 1591, 1592, 1593, 1594, 1595, 1596, 1596, 1597, 1597, 1598, 1598, 1599, 1599, 1600, 1601, 1602, 1603, 1604, 1605, 1606, 1606, 1607, 1607, 1608, 1608, 1609, 1609, 1610, 1611, 1612, 1613, 1614, 1615, 1616, 1617, 1617, 1618, 1618, 1619, 1619, 1620, 1621, 1622, 1623, 1624, 1625, 1626, 1627, 1627, 1628, 1628, 1629, 1629, 1630, 1631, 1632, 1633, 1634, 1635, 1636, 1637, 1637, 1638, 1638, 1639, 1639, 1640, 1641, 1642, 1643, 1644, 1645, 1646, 1647, 1647, 1648, 1648, 1649, 1649, 1650, 1651, 1652, 1653, 1654, 1655, 1656, 1657, 1657, 1658, 1658, 1659, 1659, 1660, 1661, 1662, 1663, 1664, 1665, 1666, 1667, 1667, 1668, 1668, 1669, 1669, 1670, 1671, 1672, 1673, 1674, 1675, 1676, 1677, 1677, 1678, 1678, 1679, 1679, 1680, 1681, 1682, 1683, 1684, 1685, 1686, 1687, 1687, 1688, 1688, 1689, 1689, 1690, 1691, 1692, 1693, 1694, 1695, 1696, 1696, 1697, 1697, 1698, 1698, 1699, 1699, 1700, 1701, 1702, 1703, 1704, 1705, 1706, 1706, 1707, 1707, 1708, 1708, 1709, 1709, 1710, 1711, 1712, 1713, 1714, 1715, 1716, 1717, 1717, 1718, 1718, 1719, 1719, 1720, 1721, 1722, 1723, 1724, 1725, 1726, 1727, 1727, 1728, 1728, 1729, 1729, 1730, 1731, 1732, 1733, 1734, 1735, 1736, 1737, 1737, 1738, 1738, 1739, 1739, 1740, 1741, 1742, 1743, 1744
```

```

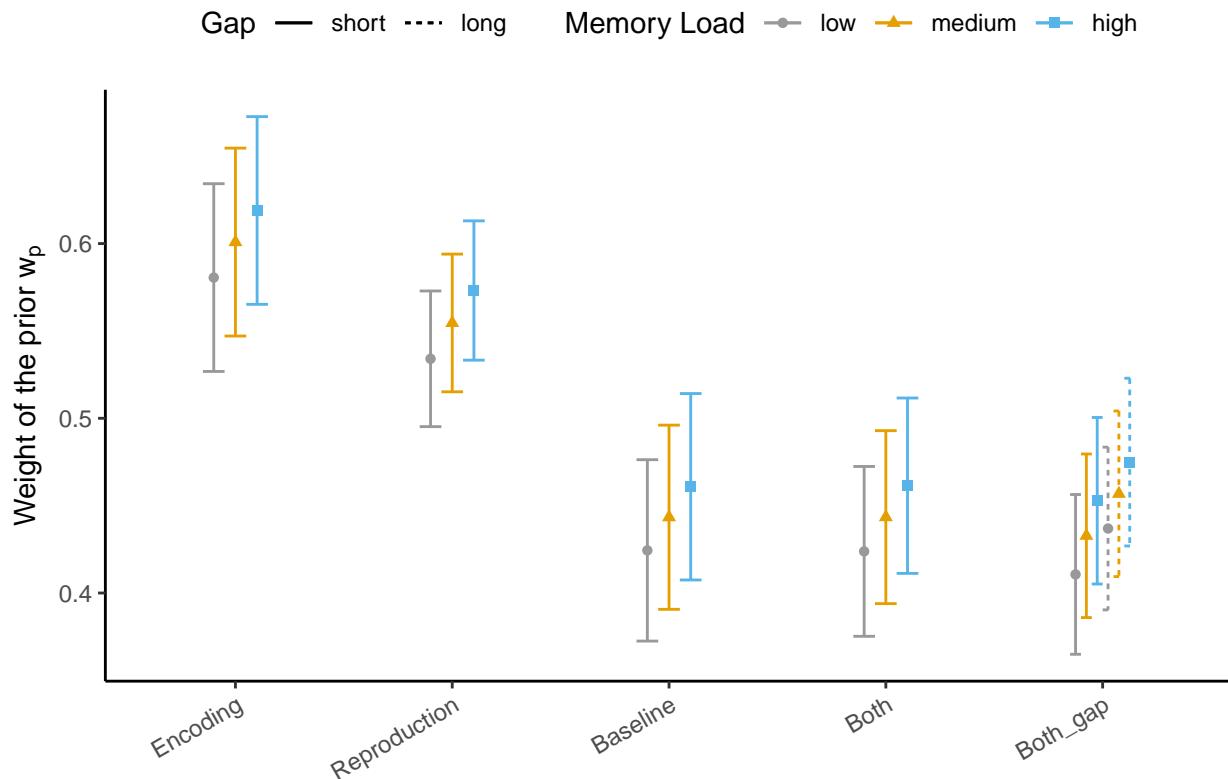
x = "",
y = TeX("Weight of the prior $w_p$"),
color = "Memory Load",
shape = "Memory Load",
linetype = "Gap"
) +
theme_new+scale_x_discrete(guide = guide_axis(n.dodge = 1)) +
theme(axis.text.x = element_text(angle = 30, hjust = 1))+theme(legend.position = "top",
legend.direction = "horizontal")

```

`summarise()` has grouped output by 'Exp', 'WMSize'. You can override using the
`groups` argument.

plt_wp_Gap

`geom_line()`: Each group consists of only one observation.
i Do you need to adjust the group aesthetic?



ggsave(paste0(getwd(), "/figures/plt_wp_Gap.png"), plt_wp_Gap, width = 7, height = 4)

`geom_line()`: Each group consists of only one observation.
i Do you need to adjust the group aesthetic?

```

# ----
colorSet5_bw <- list(
  scale_color_manual(values = c("black", "grey20", "grey45")),
  scale_linetype_manual(values = c("solid", "dotted"))
)

```

```

plt_wp_Gap_wb <- ggplot(
  data = ParaList %>%

```

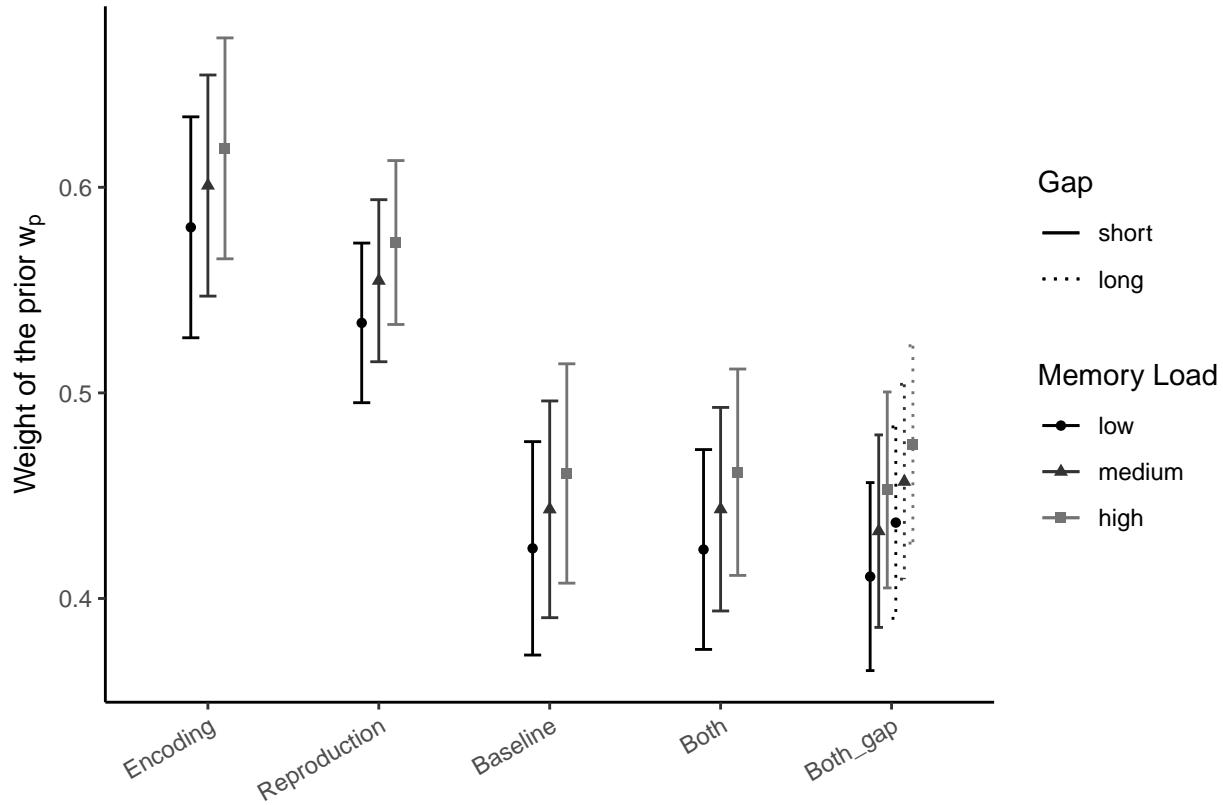
```

dplyr::group_by(Exp, WMSize, Gap) %>%
dplyr::summarise(
  m_wp = mean(w_p),
  n = n(),
  se_wp = sd(w_p) / sqrt(n - 1),
  .groups = "drop"
),
aes(
  x = Exp,
  y = m_wp,
  ymin = m_wp - se_wp,
  ymax = m_wp + se_wp,
  group = interaction(Exp, WMSize, Gap),
  color = factor(WMSize),
  shape = factor(WMSize),
  linetype = factor(Gap)
)
) +
  geom_line(position = position_dodge(width = 0.3)) +
  geom_point(position = position_dodge(width = 0.3)) +
  geom_errorbar(width = .3, position = position_dodge(width = .3)) +
  colorSet5_bw +
  scale_x_discrete(labels = exp_labels) +
  labs(
    x = "",
    y = TeX("Weight of the prior $w_p$"),
    color = "Memory Load",
    shape = "Memory Load",
    linetype = "Gap"
) +
  theme_new +
  theme(legend.position="top") +
  theme_new +
  scale_x_discrete(guide = guide_axis(n.dodge = 1)) +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))

## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
plt_wp_Gap_wb

## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?

```



predicted mean bias

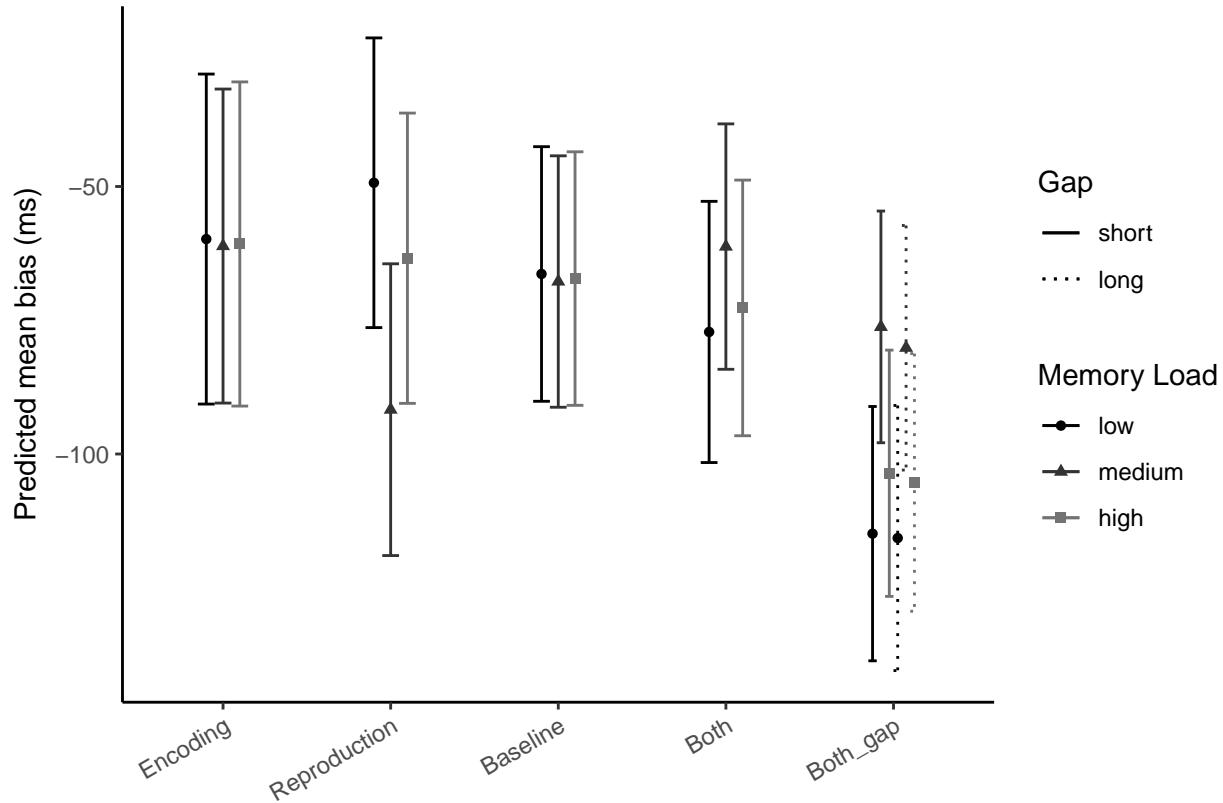
```

plt_pred_Bias_bw <- ggplot(data = mdat_all %>% dplyr::group_by(Exp, WMSize, Gap) %>% dplyr::summarise(mmP
  geom_line(stat = "identity", position = position_dodge(width = 0.3))+
  geom_point(stat = "identity", position = position_dodge(width = 0.3))+ 
  geom_errorbar(width=.3, position = position_dodge(width = .3)) +
  scale_x_discrete(labels = exp_labels) +
  colorSet5_bw+
  labs(x = "", y = TeX("Predicted mean bias (ms)"), color = 'Memory Load', shape = 'Memory Load', linety
  theme_new+scale_x_discrete(guide = guide_axis(n.dodge = 1)) +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))

## `summarise()` has grouped output by 'Exp', 'WMSize'. You can override using the
## `groups` argument.
## Scale for x is already present. Adding another scale for x, which will replace
## the existing scale.
plt_pred_Bias_bw

## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?

```



```

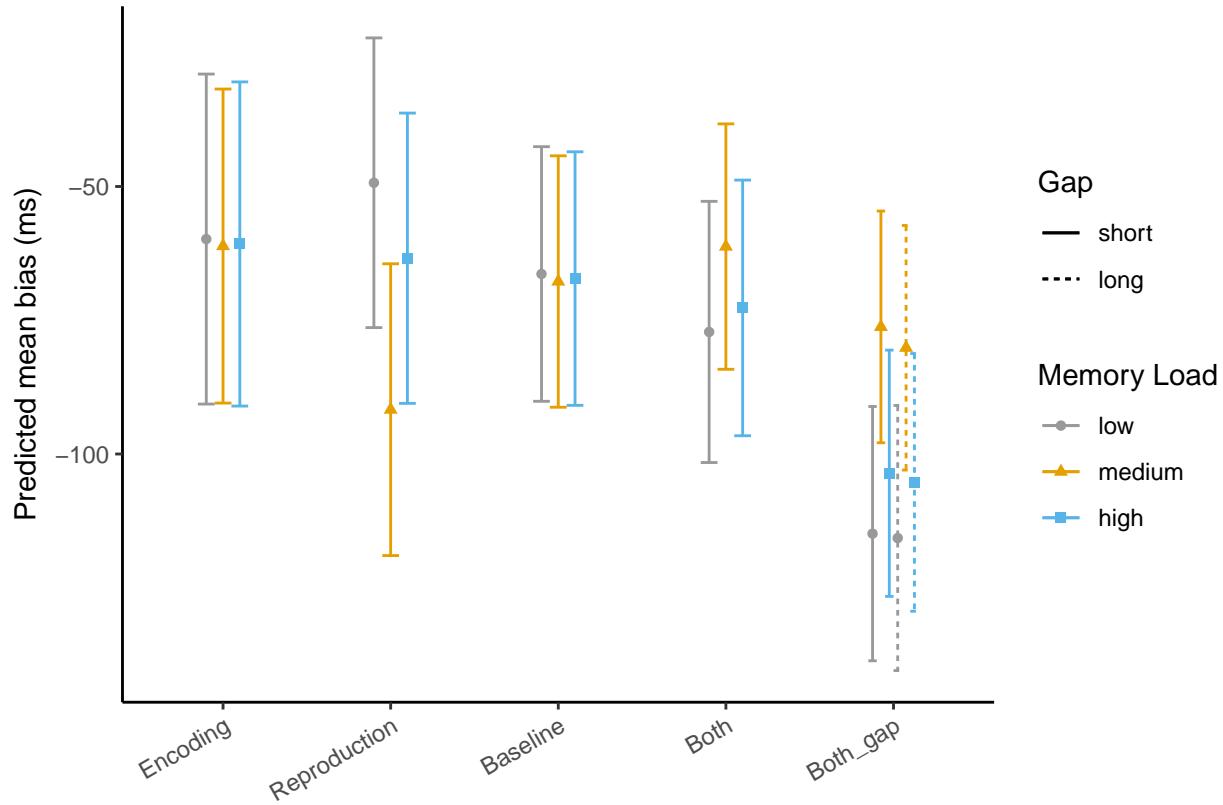
plt_pred_Bias <- ggplot(data = mdat_all %>% dplyr::group_by(Exp, WMSize, Gap) %>% dplyr::summarise(mmPred
  geom_line(stat = "identity", position = position_dodge(width = 0.3))+
  geom_point(stat = "identity", position = position_dodge(width = 0.3))+
  geom_errorbar(width=.3, position = position_dodge(width = .3)) +
  scale_x_discrete(labels = exp_labels) +
  colorSet5+
  labs(x = "", y = TeX("Predicted mean bias (ms)"), color = 'Memory Load', shape = 'Memory Load', linetyp
    theme_new+scale_x_discrete(guide = guide_axis(n.dodge = 1)) +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))

## `summarise()` has grouped output by 'Exp', 'WMSize'. You can override using the
## `.groups` argument.
## Scale for x is already present. Adding another scale for x, which will replace
## the existing scale.

plt_pred_Bias

## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?

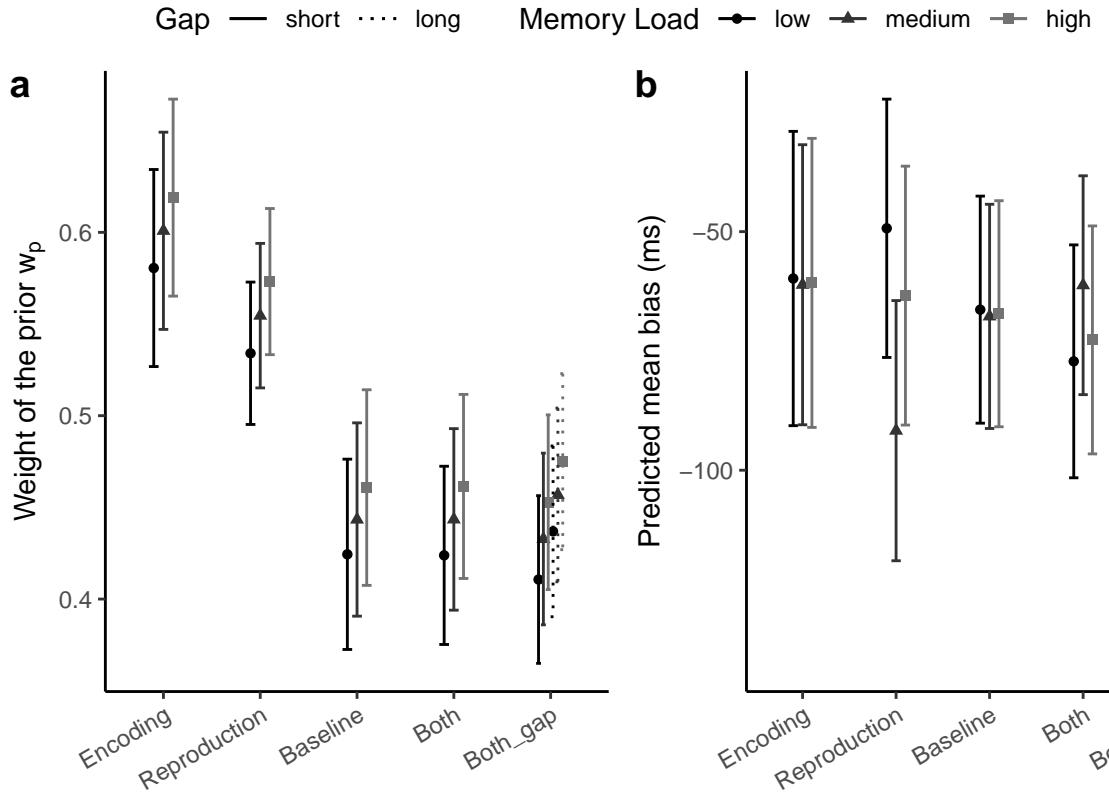
```



```

fig7_wb<-ggarrange(plt_wp_Gap_wb, plt_pred_Bias_bw, common.legend = TRUE, ncol=2, nrow=1, labels = c("a", "b"))
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
ggsave(paste0(getwd(), "/figures/fig7_wb_", modelname, ".png")), fig7_wb, width = 7, height = 4)
fig7_wb

```



```

fig7<-ggarrange(plt_wp_Gap, plt_pred_Bias, common.legend = TRUE, ncol=2, nrow=1, labels = c("a", "b"))

## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
ggsave(paste0(getwd(), "/figures/fig7_",modelname,".png"), fig7, width = 7, height = 4)
fig7

```

Gap — short ---- long Memory Load ● low ▲ medium ■ high

