Image Inference (Experiment 4)

Preprocessing

```r
# Read in the participant data.
data_0 = read_csv(file.path(human_path, "raw_data.csv"), quote="~")

# Convert the JSON string into JSON.
data_1 = lapply(data_0$data, fromJSON)

# Extract the trial information for each participant and stack them.
age = c()
data_3 = tibble()
for (p in 1:length(data_1)) {
  # Trim the map and add the participant ID back in.
  data_2 = data_1[p][[1]]$trials %>%
    as.data.frame() %>%
    mutate(map=gsub(".png", "", map),
           prior=gsub("^.*_doors", "doors", prior),
           prior=gsub(".png", "", prior),
           unique_id=data_1[p][[1]]$id,
           quiz_attempts=data_1[p][[1]]$catch_trials$quiz_attempts)

  # Extract and store this participant's age.
  age = c(age, as.integer(data_1[p][[1]]$subject_information$age))

  # Stack the trial information for the current participant.
  data_3 = rbind(data_3, data_2)
}

# Write the preprocessed data.
write_csv(data_3, file.path(human_path, "data.csv"))

# Combine the posterior over goals for each of the maps.
model_1 = tibble()
goal_files = list.files(model_path, pattern="_goals_")
for (goal_file in goal_files) {
  # Read in the goal predictions for this map and do some preprocessing.
  model_0 = read_csv(file.path(model_path, goal_file)) %>%
    mutate(map=gsub("_goals_posterior.csv", "", goal_file),
           goal=ifelse(goal=="Blue", "A",
                       ifelse(goal=="Orange", "B", "C")))

  # Stack these goal predictions.
  model_1 = model_1 %>%
    rbind(model_0)
```

```
}

# Write the goal predictions.
write_csv(model_1, file.path(model_path, "goal_predictions.csv"))

# Combine the posterior over entrances for each of the maps.
model_5 = tibble()
state_files = list.files(model_path, pattern="_state")
for (state_file in state_files) {
  # Read in the path predictions.
  model_2 = read_csv(file.path(model_path, state_file))

  # Do some preprocessing.
  model_3 = model_2 %>%
    rownames_to_column("path") %>%
    gather(step, state, names(model_2)[grepl("s_", names(model_2))]) %>%
    separate(step, into=c("temp", "time"), sep="_") %>%
    mutate(path=as.numeric(path)-1, time=as.numeric(time),
           state=as.numeric(state)) %>%
    arrange(path, time) %>%
    mutate(x=state%%map_width+1, y=ceiling(state/map_width)) %>%
    select(-temp, -map_height, -map_width)

  # Extract the the first state from each path (i.e., the entrance).
  model_4 = model_3 %>%
    filter(time==0) %>%
    group_by(state) %>%
    summarize(probability=sum(probability)) %>%
    rename(entrance=state) %>%
    mutate(map=gsub("_states_posterior.csv", "", state_file))

  # Stack these entrance predictions.
  model_5 = model_5 %>%
    rbind(model_4)
}

# Write the entrance predictions.
write_csv(model_5, file.path(model_path, "entrance_predictions.csv"))
```

Goal Prior Manipulation

First, we'll analyze the trials where we manipulated each participant's prior over the goals.

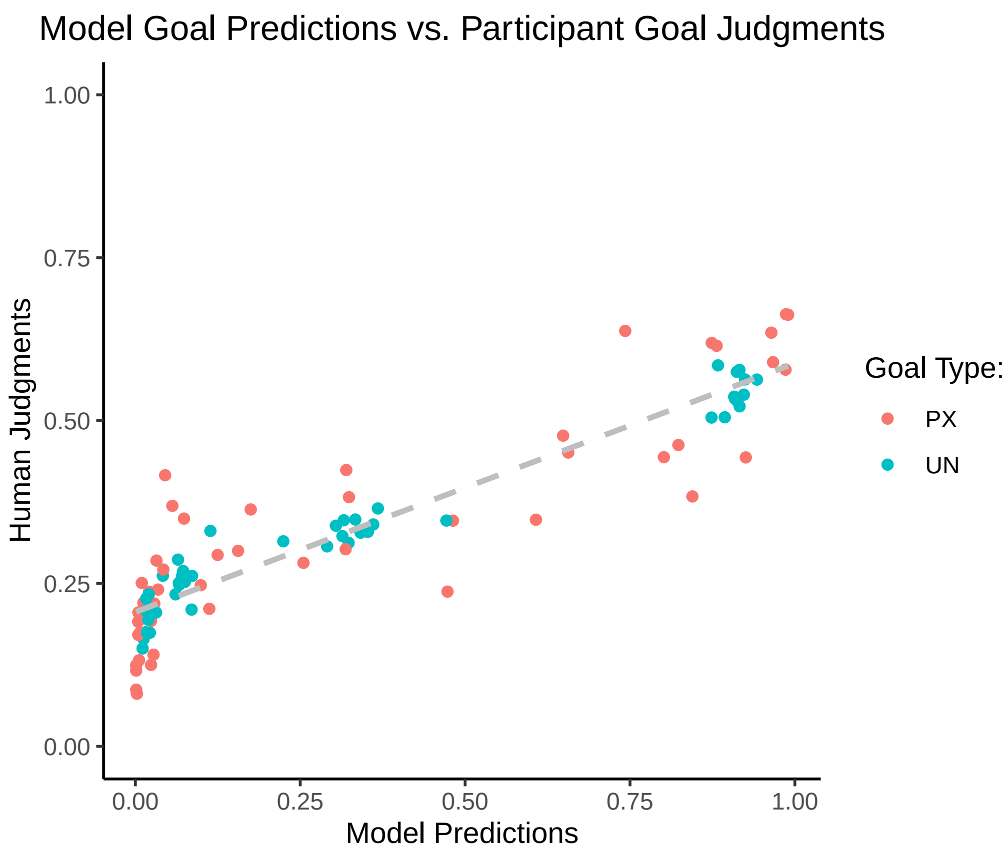
Goal Inference

Here we generate a scatter plot comparing participant judgments (N=160; M=33.49 years, SD=11.36 years) against our model predictions on the goal inferences.

```
# # Read in the preprocessed data.
data_3 = read_csv(file.path(human_path, "data.csv"))
```

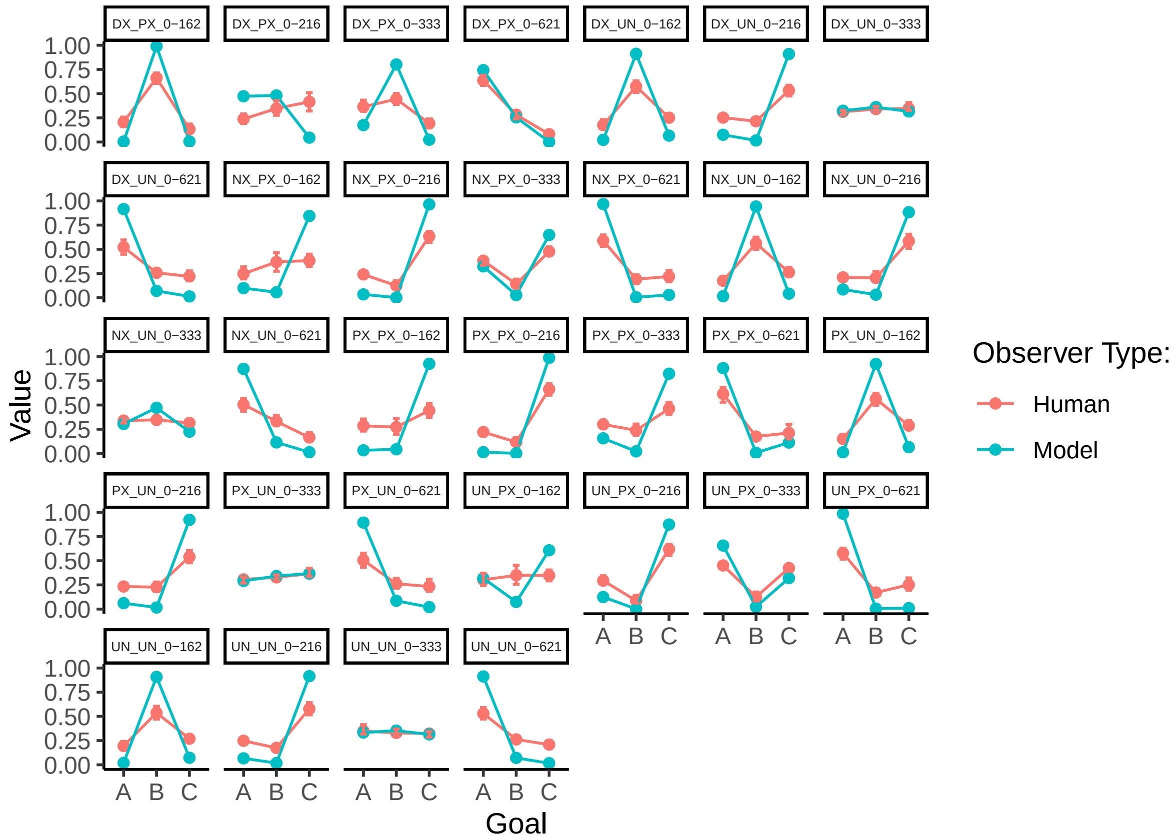
```r
# Select and normalize the goal judgments.
data_4 = data_3 %>%
  filter(grepl("goals", prior)) %>%
  unite("map", map:prior, sep="_") %>%
  select(unique_id, map, A, B, C) %>%
  gather(goal, human, A, B, C) %>%
  left_join(do(., summarize(group_by(., unique_id, map),
                            total_human=sum(human)))) %>%
  mutate(human=human/total_human) %>%
  select(-total_human)

# Define the bootstrap function for the bootstrap statistic.
compute_mean = function(data, indices) {
  return(mean(data[indices]))
}

# Define the bootstrap function to simulate the data.
compute_bootstrap = function(data) {
  # Run the simulations.
  simulations = boot(data=data,
                     statistic=compute_mean,
                     R=10000)

  return(boot.ci(simulations, type="bca")$bca)
}

# Compute the bootstrapped 95% CIs.
set.seed(seed)
ci = data.frame()
for (m in unique(data_4$map)) {
  # Filter the current map.
  data_5 = data_4 %>%
    filter(map==m)

  # Compute the bootstrap for each dependent measure.
  bootstrap_A = compute_bootstrap(filter(data_5, goal=="A")$human)
  bootstrap_B = compute_bootstrap(filter(data_5, goal=="B")$human)
  bootstrap_C = compute_bootstrap(filter(data_5, goal=="C")$human)

  # Store the bootstrapped 95% CIs for this pair.
  ci = rbind(ci, data.frame(map=rep(m, 3),
                            goal=c("A", "B", "C"),
                            lower=c(bootstrap_A[4],
                                    bootstrap_B[4],
                                    bootstrap_C[4]),
                            upper=c(bootstrap_A[5],
                                    bootstrap_B[5],
                                    bootstrap_C[5])))
}

# Read in the goal predictions.
model_6 = read_csv(file.path(model_path, "goal_predictions.csv"))
```

```r
# Perform some basic preprocessing and then z-score the model predictions.
model_7 = model_6 %>%
  rename(model=probability) %>%
  mutate(z_model=scale(model)[,1])

# z-score the participant judgments and then merge them with the bootstrapped
# 95% CIs and the model predictions.
data_6 = data_4 %>%
  group_by(unique_id) %>%
  mutate(z_human=scale(human)[,1]) %>%
  ungroup() %>%
  group_by(map, goal) %>%
  summarize(mean_human=mean(human), mean_z_human=mean(z_human)) %>%
  ungroup() %>%
  left_join(ci) %>%
  left_join(model_7)

# Plot the goal comparison.
plot_0 = data_6 %>%
  ggplot(aes(x=model, y=mean_human, label=map)) +
  geom_point(aes(color=substr(map, 4, 5))) +
  geom_smooth(method="lm", se=FALSE, linetype="dashed", color="grey") +
  ggtitle("Model Goal Predictions vs. Participant Goal Judgments") +
  xlab("Model Predictions") +
  ylab("Human Judgments") +
  ylim(0.0, 1.0) +
  scale_color_discrete(name="Goal Type:") +
  theme_classic() +
  theme(aspect.ratio=1.0,
        plot.title=element_text(hjust=0.5),
        legend.title=element_text(hjust=0.5))
plot_0
```



For the goal inferences under varying goal priors, the Pearson correlation is r=0.92 (95% CI: 0.87-0.94). Next, we generate goal inference comparisons for each trial.

```
# Plot goal inferences by trial.
plot_1 = data_6 %>%
  mutate(map=gsub("_goals", "", map)) %>%
  gather(type, value, mean_human, model) %>%
  ggplot(aes(x=goal, y=value, group=type)) +
  geom_point(aes(color=type)) +
  geom_line(aes(color=type)) +
  geom_errorbar(aes(ymin=lower, ymax=upper), color="#F8766D", width=0.2) +
  facet_wrap(~map, ncol=7) +
  xlab("Goal") +
  ylab("Value") +
  scale_color_discrete(name="Observer Type:",
                       limits=c("mean_human", "model"),
                       labels=c("Human", "Model")) +
  theme_classic() +
  theme(aspect.ratio=1.0,
        legend.title=element_text(hjust=0.5),
        strip.text=element_text(size=5))
plot_1
```



Entrance Inference

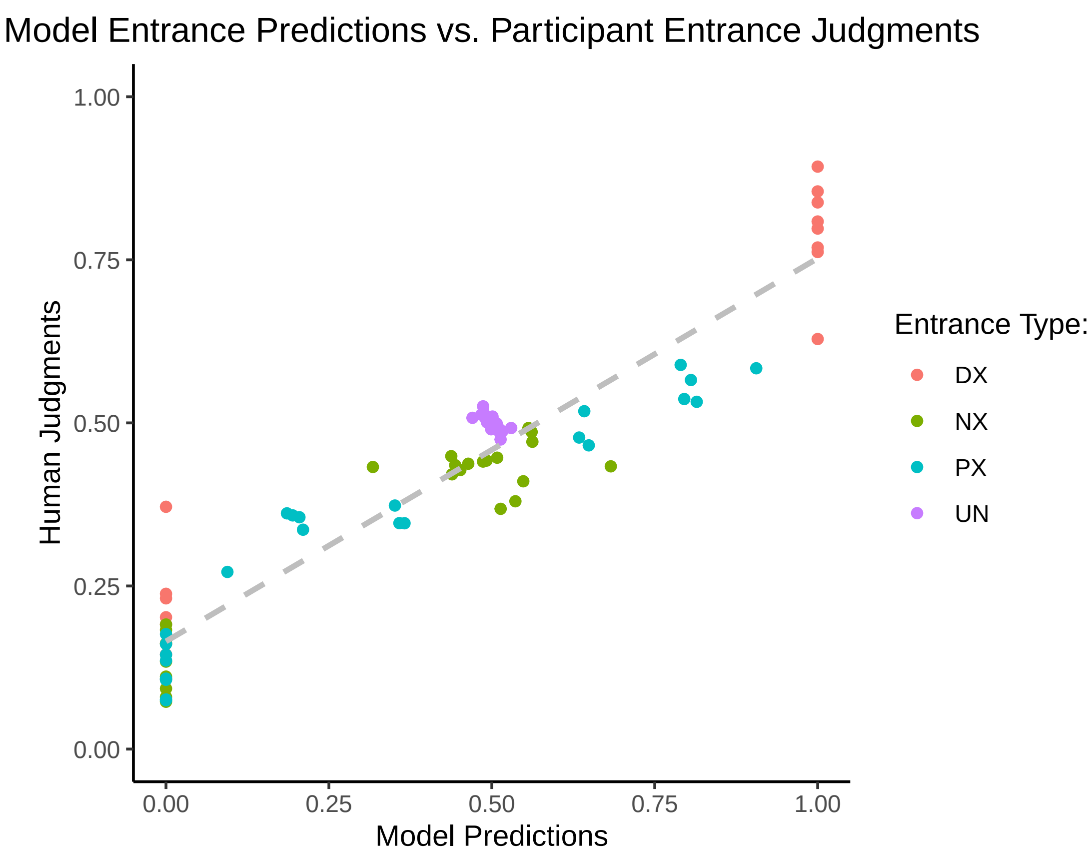
Here we generate a scatter plot comparing participant judgments (N=160; M=33.49 years, SD=11.36 years) against our model predictions on the entrance inferences.

```
# Select and normalize the entrance judgments.
data_7 = data_3 %>%
  filter(grepl("goals", prior)) %>%
  unite("map", map:prior, sep="_") %>%
  select(unique_id, map, `1`, `2`, `3`) %>%
  gather(entrance, human, `1`, `2`, `3`) %>%
  filter(human!=-1) %>%
  left_join(do(., summarize(group_by(., unique_id, map),
                            total_human=sum(human)))) %>%
  mutate(human=human/total_human) %>%
  select(-total_human)

# Define the bootstrap function for the bootstrap statistic.
compute_mean = function(data, indices) {
  return(mean(data[indices]))
}

# Define the bootstrap function to simulate the data.
compute_bootstrap = function(data) {
  # Run the simulations.
```

```r
  simulations = boot(data=data,
                     statistic=compute_mean,
                     R=10000)

  return(boot.ci(simulations, type="bca")$bca)
}

# Compute the bootstrapped 95% CIs.
set.seed(seed)
ci = data.frame()
for (m in unique(data_7$map)) {
  # Filter the current map.
  data_8 = data_7 %>%
    filter(map==m)

  # Compute the bootstrap for each dependent measure.
  bootstrap_1 = compute_bootstrap(filter(data_8, entrance=="1")$human)
  bootstrap_2 = compute_bootstrap(filter(data_8, entrance=="2")$human)
  if (length(unique(data_8$entrance)) == 3) {
    bootstrap_3 = compute_bootstrap(filter(data_8, entrance=="3")$human)
  }

  # Store the bootstrapped 95% CIs for this pair.
  if (length(unique(data_8$entrance)) == 3) {
    ci = rbind(ci, data.frame(map=rep(m, 3),
                              entrance=c("1", "2", "3"),
                              lower=c(bootstrap_1[4],
                                      bootstrap_2[4],
                                      bootstrap_3[4]),
                              upper=c(bootstrap_1[5],
                                      bootstrap_2[5],
                                      bootstrap_3[5])))
  }
  else {
    ci = rbind(ci, data.frame(map=rep(m, 2),
                              entrance=c("1", "2"),
                              lower=c(bootstrap_1[4],
                                      bootstrap_2[4]),
                              upper=c(bootstrap_1[5],
                                      bootstrap_2[5])))
  }
}

# Read in the entrance predictions.
model_8 = read_csv(file.path(model_path, "entrance_predictions.csv"))

# Preprocess the entrance predictions.
model_9 = model_8 %>%
  separate(map, into=c("map", "prior"), sep=7) %>%
  mutate(prior=gsub("^_", "", prior))

# Read in the entrance mapping.
model_10 = read_csv(file.path(model_path, "entrance_mapping.csv"))
```

```r
# Merge the entrance predictions with the entrance mapping.
model_14 = tibble()
for (m in unique(model_9$map)) {
  # Select the current map.
  model_11 = model_9 %>%
    filter(map==m)

  # Iterate through the priors of the current map.
  for (p in unique(model_11$prior)) {
    # Select the relevant entrance mapping and append the current prior.
    model_12 = model_10 %>%
      filter(map==m) %>%
      mutate(prior=p)

    # Apply the entrance mapping to the current map.
    model_13 = model_9 %>%
      filter(map==m, prior==p) %>%
      right_join(model_12) %>%
      mutate(probability=ifelse(is.na(probability), 0, probability)) %>%
      select(-entrance) %>%
      rename(entrance=number, model=probability) %>%
      mutate(entrance=as.character(entrance), z_model=scale(model)[,1]) %>%
      unite("map", map:prior, sep="_")

    # Add the current map to the stack.
    model_14 = rbind(model_14, model_13)
  }
}

# z-score the participant judgments and then merge them with the bootstrapped
# 95% CIs and the model predictions.
data_9 = data_7 %>%
  group_by(unique_id) %>%
  mutate(z_human=scale(human)[,1]) %>%
  ungroup() %>%
  group_by(map, entrance) %>%
  summarize(mean_human=mean(human), mean_z_human=mean(z_human)) %>%
  ungroup() %>%
  left_join(ci) %>%
  left_join(model_14)

# Plot the entrance comparison.
plot_2 = data_9 %>%
  ggplot(aes(x=model, y=mean_human, label=map)) +
  geom_point(aes(color=substr(map, 0, 2))) +
  geom_smooth(method="lm", se=FALSE, linetype="dashed", color="grey") +
  ggtitle("Model Entrance Predictions vs. Participant Entrance Judgments") +
  xlab("Model Predictions") +
  ylab("Human Judgments") +
  ylim(0.0, 1.0) +
  scale_color_discrete(name="Entrance Type:") +
  theme_classic() +
  theme(aspect.ratio=1.0,
```

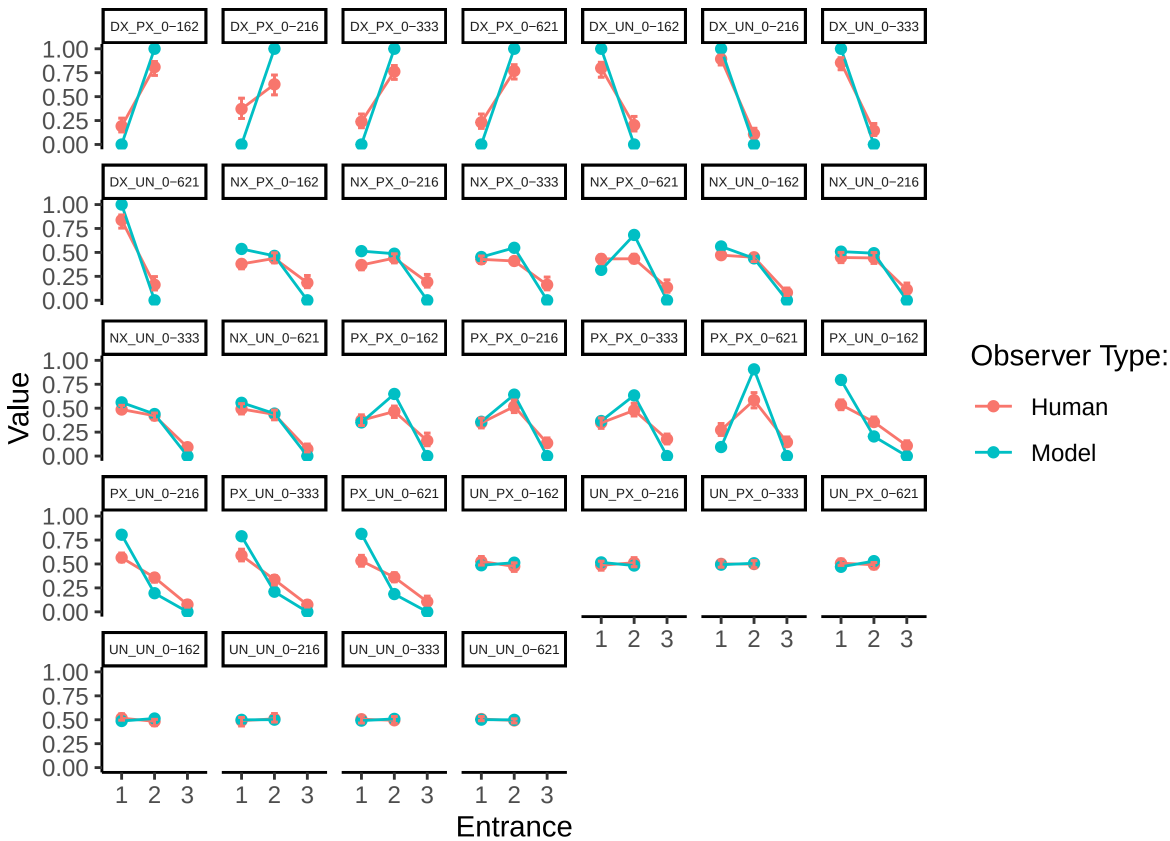
```
        plot.title=element_text(hjust=0.5),
        legend.title=element_text(hjust=0.5))
plot_2
```



For the entrance inferences under varying goal priors, the Pearson correlation is r=0.95 (95% CI: 0.92-0.96). Next, we generate entrance inference comparisons for each trial.

```
# Plot entrance inferences by trial.
plot_3 = data_9 %>%
  mutate(map=gsub("_goals", "", map)) %>%
  gather(type, value, mean_human, model) %>%
  ggplot(aes(x=entrance, y=value, group=type)) +
  geom_point(aes(color=type)) +
  geom_line(aes(color=type)) +
  geom_errorbar(aes(ymin=lower, ymax=upper), color="#F8766D", width=0.2) +
  facet_wrap(~map, ncol=7) +
  xlab("Entrance") +
  ylab("Value") +
  scale_color_discrete(name="Observer Type:",
                       limits=c("mean_human", "model"),
                       labels=c("Human", "Model")) +
  theme_classic() +
  theme(aspect.ratio=1.0,
        legend.title=element_text(hjust=0.5),
```

```
        strip.text=element_text(size=5))
plot_3
```

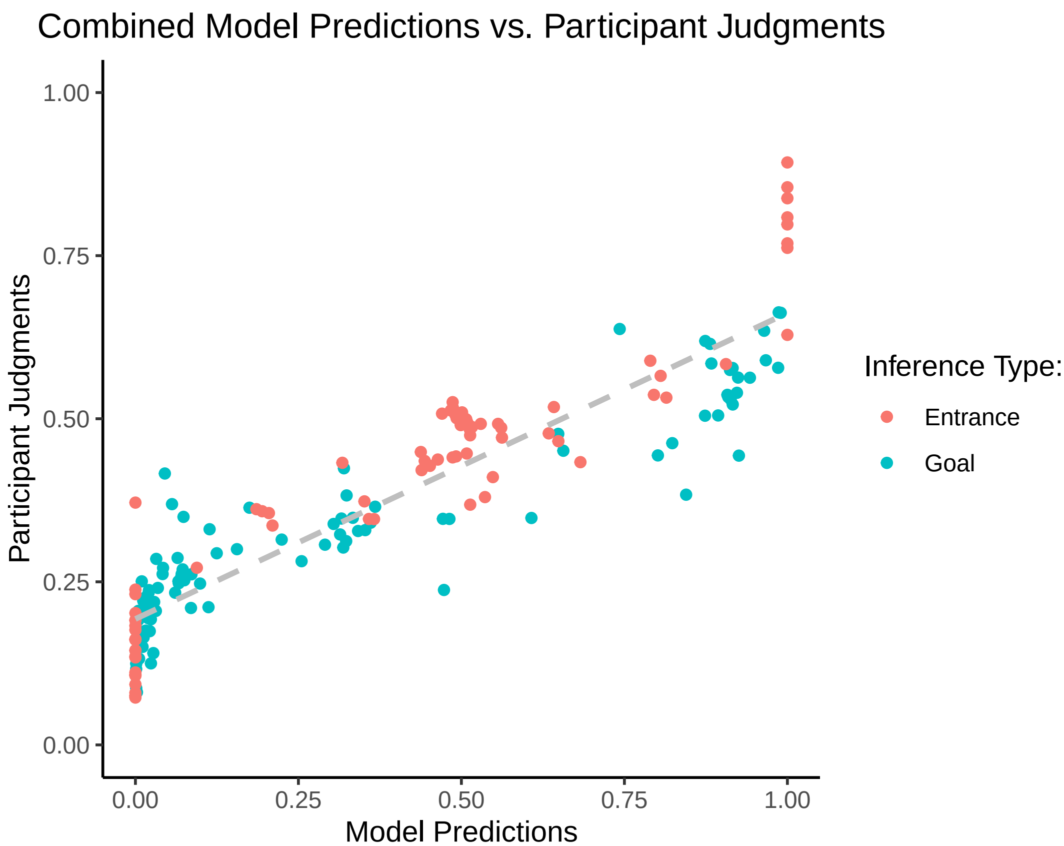


Combined Inference

Now we can combine both scatter plots.

```
# Merge the two inferences.
data_10 = data_6 %>%
  mutate(type="goal", inference=goal) %>%
  select(-goal) %>%
  rbind(select(mutate(data_9, type="entrance", inference=entrance), -entrance))

plot_4 = data_10 %>%
  ggplot(aes(x=model, y=mean_human, label=map)) +
  geom_point(aes(color=type)) +
  geom_smooth(method="lm", se=FALSE, linetype="dashed", color="grey") +
  ggtitle("Combined Model Predictions vs. Participant Judgments") +
  xlab("Model Predictions") +
  ylab("Participant Judgments") +
  ylim(0.0, 1.0) +
  scale_color_discrete(name="Inference Type:",
                       limits=c("entrance", "goal"),
                       labels=c("Entrance", "Goal")) +
```

```
  theme_classic() +
  theme(aspect.ratio=1.0,
        plot.title=element_text(hjust=0.5),
        legend.title=element_text(hjust=0.5))
plot_4
```



The combined Pearson correlation under varying goal priors is r=0.91 (95% CI: 0.89-0.93).

Entrance Prior Manipulation

Next, we'll analyze the trials where we manipulated each participant's prior over the entrances.

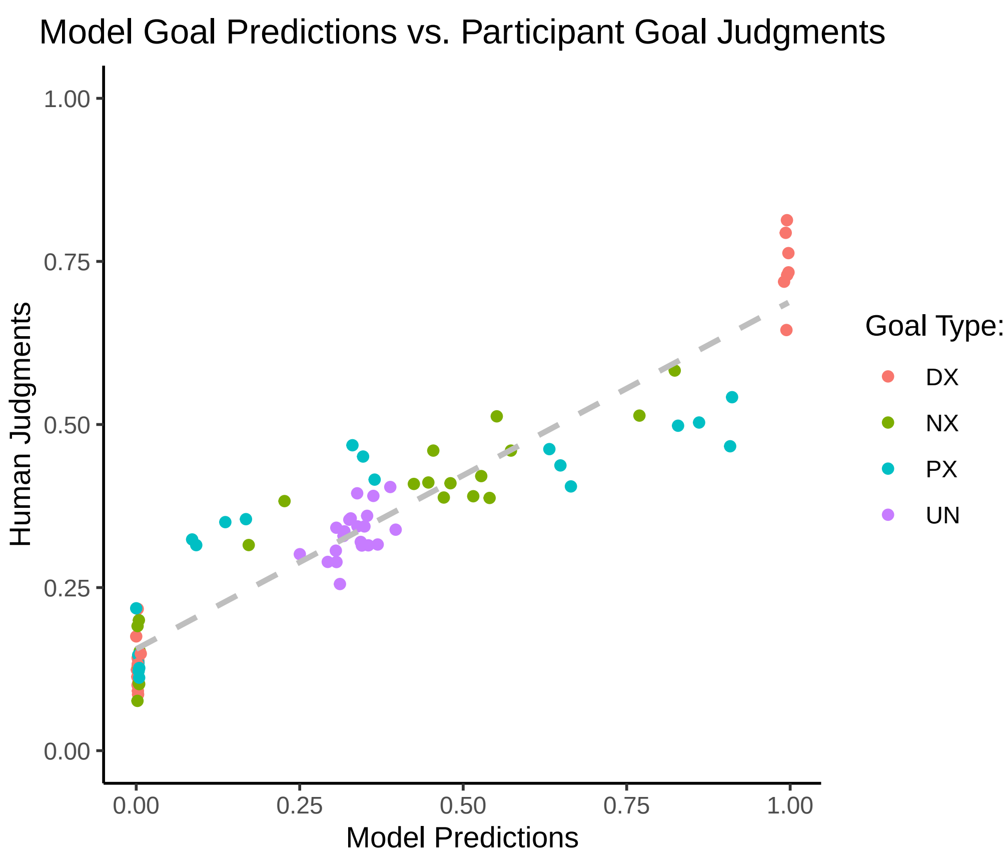
Goal Inference

Here we generate a scatter plot comparing participant judgments (N=160; M=33.49 years, SD=11.36 years) against our model predictions on the goal inferences.

```
# Select and normalize the goal judgments.
data_11 = data_3 %>%
  filter(grepl("doors", prior)) %>%
  unite("map", map:prior, sep="_") %>%
  select(unique_id, map, A, B, C) %>%
```

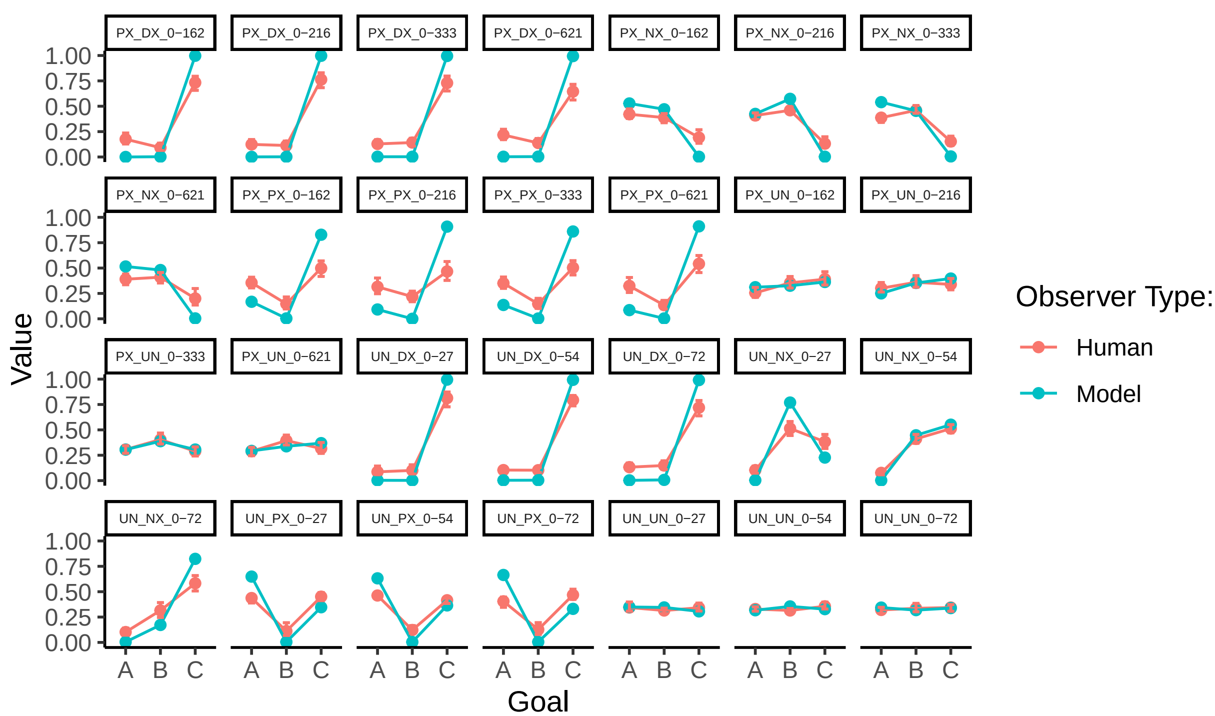
```r
  gather(goal, human, A, B, C) %>%
  left_join(do(., summarize(group_by(., unique_id, map),
                            total_human=sum(human)))) %>%
  mutate(human=human/total_human) %>%
  select(-total_human)

# Define the bootstrap function for the bootstrap statistic.
compute_mean = function(data, indices) {
  return(mean(data[indices]))
}

# Define the bootstrap function to simulate the data.
compute_bootstrap = function(data) {
  # Run the simulations.
  simulations = boot(data=data,
                     statistic=compute_mean,
                     R=10000)

  return(boot.ci(simulations, type="bca")$bca)
}

# Compute the bootstrapped 95% CIs.
set.seed(seed)
ci = data.frame()
for (m in unique(data_11$map)) {
  # Filter the current map.
  data_12 = data_11 %>%
    filter(map==m)

  # Compute the bootstrap for each dependent measure.
  bootstrap_A = compute_bootstrap(filter(data_12, goal=="A")$human)
  bootstrap_B = compute_bootstrap(filter(data_12, goal=="B")$human)
  bootstrap_C = compute_bootstrap(filter(data_12, goal=="C")$human)

  # Store the bootstrapped 95% CIs for this pair.
  ci = rbind(ci, data.frame(map=rep(m, 3),
                            goal=c("A", "B", "C"),
                            lower=c(bootstrap_A[4],
                                    bootstrap_B[4],
                                    bootstrap_C[4]),
                            upper=c(bootstrap_A[5],
                                    bootstrap_B[5],
                                    bootstrap_C[5])))
}

# z-score the participant judgments and then merge them with the bootstrapped
# 95% CIs and the model predictions.
data_13 = data_11 %>%
  group_by(unique_id) %>%
  mutate(z_human=scale(human)[,1]) %>%
  ungroup() %>%
  group_by(map, goal) %>%
  summarize(mean_human=mean(human), mean_z_human=mean(z_human)) %>%
```

```
  ungroup() %>%
  left_join(ci) %>%
  left_join(model_7)

# Plot the goal comparison.
plot_5 = data_13 %>%
  ggplot(aes(x=model, y=mean_human, label=map)) +
  geom_point(aes(color=substr(map, 4, 5))) +
  geom_smooth(method="lm", se=FALSE, linetype="dashed", color="grey") +
  ggtitle("Model Goal Predictions vs. Participant Goal Judgments") +
  xlab("Model Predictions") +
  ylab("Human Judgments") +
  ylim(0.0, 1.0) +
  scale_color_discrete(name="Goal Type:") +
  theme_classic() +
  theme(aspect.ratio=1.0,
        plot.title=element_text(hjust=0.5),
        legend.title=element_text(hjust=0.5))
plot_5
```



For the goal inferences under varying entrance priors, the Pearson correlation is r=0.94 (95% CI: 0.91-0.96). Next, we generate goal inference comparisons for each trial.

```r
# Plot goal inferences by trial.
plot_6 = data_13 %>%
  mutate(map=gsub("_doors", "", map)) %>%
  gather(type, value, mean_human, model) %>%
  ggplot(aes(x=goal, y=value, group=type)) +
  geom_point(aes(color=type)) +
  geom_line(aes(color=type)) +
  geom_errorbar(aes(ymin=lower, ymax=upper), color="#F8766D", width=0.2) +
  facet_wrap(~map, ncol=7) +
  xlab("Goal") +
  ylab("Value") +
  scale_color_discrete(name="Observer Type:",
                       limits=c("mean_human", "model"),
                       labels=c("Human", "Model")) +
  theme_classic() +
  theme(aspect.ratio=1.0,
        legend.title=element_text(hjust=0.5),
        strip.text=element_text(size=5))
plot_6
```

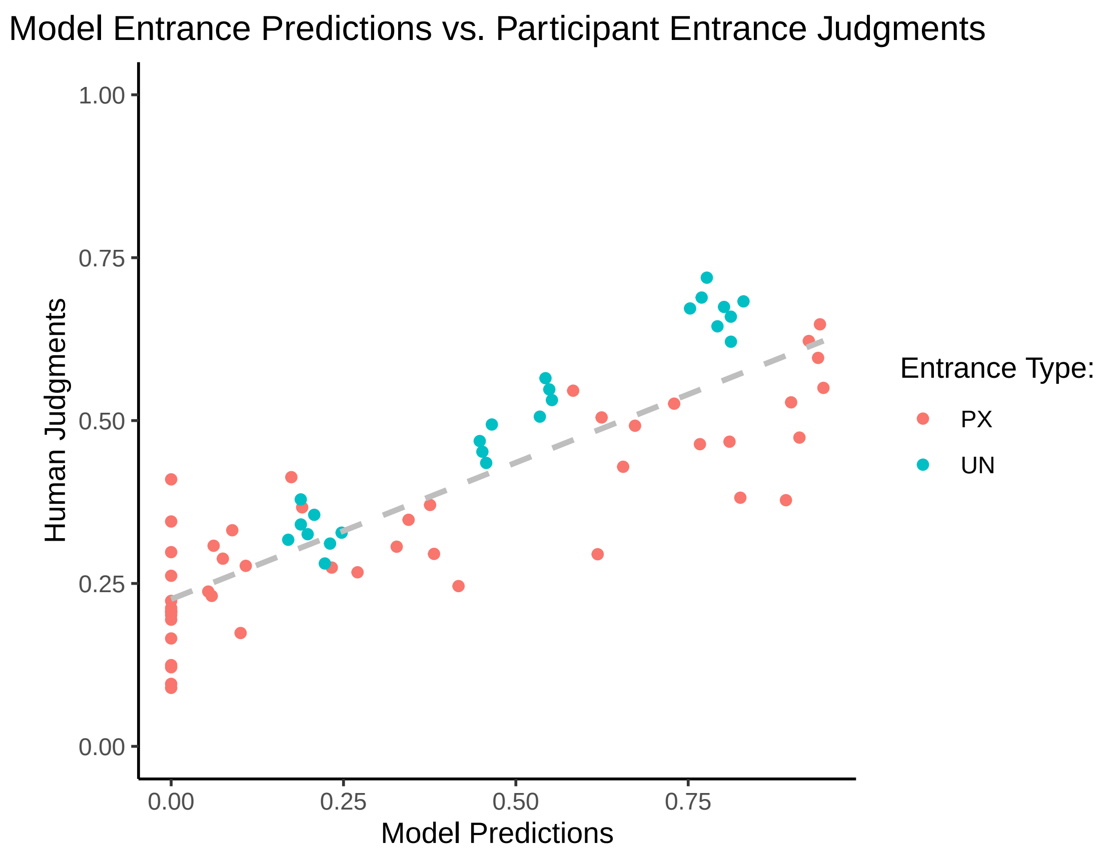


Entrance Inference

Here we generate a scatter plot comparing participant judgments (N=160; M=33.49 years, SD=11.36 years) against our model predictions on the entrance inferences.

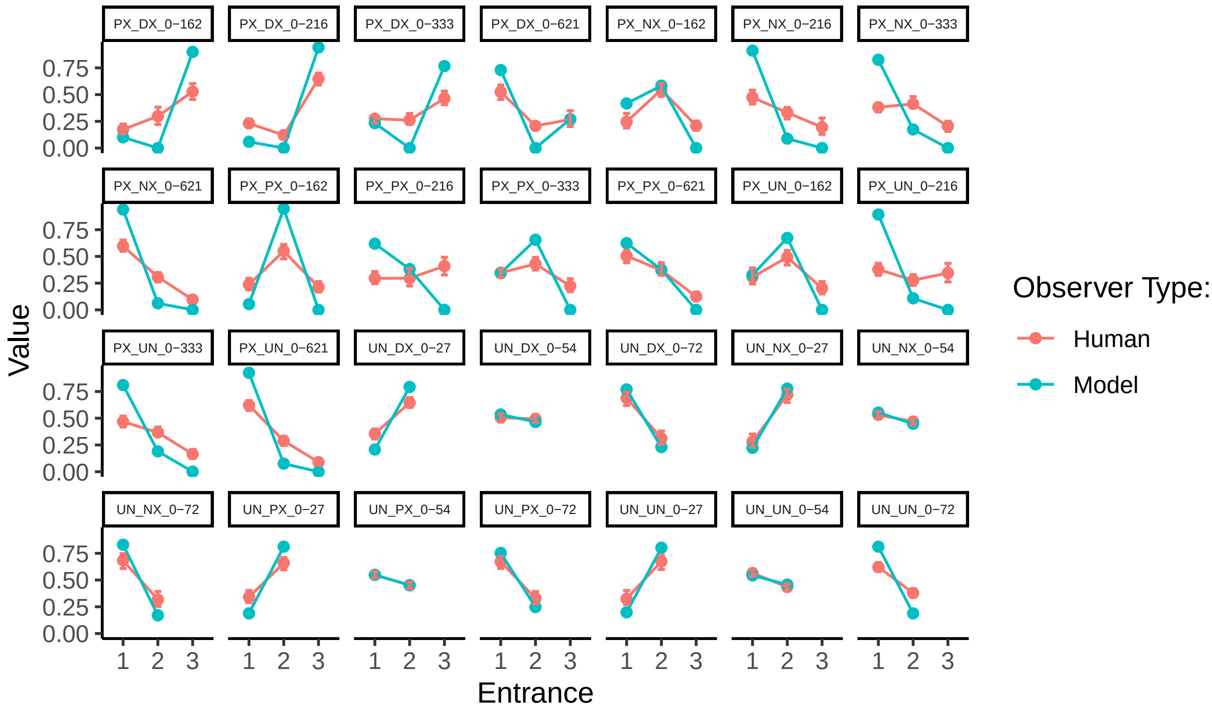
```r
# Select and normalize the entrance judgments.
data_14 = data_3 %>%
  filter(grepl("doors", prior)) %>%
  unite("map", map:prior, sep="_") %>%
  select(unique_id, map, `1`, `2`, `3`) %>%
  gather(entrance, human, `1`, `2`, `3`) %>%
  filter(human!=-1) %>%
  left_join(do(., summarize(group_by(., unique_id, map),
                            total_human=sum(human)))) %>%
  mutate(human=human/total_human) %>%
  select(-total_human)

# Define the bootstrap function for the bootstrap statistic.
compute_mean = function(data, indices) {
  return(mean(data[indices]))
}

# Define the bootstrap function to simulate the data.
compute_bootstrap = function(data) {
  # Run the simulations.
  simulations = boot(data=data,
                     statistic=compute_mean,
                     R=10000)

  return(boot.ci(simulations, type="bca")$bca)
}

# Compute the bootstrapped 95% CIs.
set.seed(seed)
ci = data.frame()
for (m in unique(data_14$map)) {
  # Filter the current map.
  data_15 = data_14 %>%
    filter(map==m)

  # Compute the bootstrap for each dependent measure.
  bootstrap_1 = compute_bootstrap(filter(data_15, entrance=="1")$human)
  bootstrap_2 = compute_bootstrap(filter(data_15, entrance=="2")$human)
  if (length(unique(data_15$entrance)) == 3) {
    bootstrap_3 = compute_bootstrap(filter(data_15, entrance=="3")$human)
  }

  # Store the bootstrapped 95% CIs for this pair.
  if (length(unique(data_15$entrance)) == 3) {
    ci = rbind(ci, data.frame(map=rep(m, 3),
                              entrance=c("1", "2", "3"),
                              lower=c(bootstrap_1[4],
                                      bootstrap_2[4],
                                      bootstrap_3[4]),
                              upper=c(bootstrap_1[5],
                                      bootstrap_2[5],
                                      bootstrap_3[5])))
  }
```

```r
  else {
    ci = rbind(ci, data.frame(map=rep(m, 2),
                              entrance=c("1", "2"),
                              lower=c(bootstrap_1[4],
                                      bootstrap_2[4]),
                              upper=c(bootstrap_1[5],
                                      bootstrap_2[5])))
  }
}

# z-score the participant judgments and then merge them with the bootstrapped
# 95% CIs and the model predictions.
data_16 = data_14 %>%
  group_by(unique_id) %>%
  mutate(z_human=scale(human)[,1]) %>%
  ungroup() %>%
  group_by(map, entrance) %>%
  summarize(mean_human=mean(human), mean_z_human=mean(z_human)) %>%
  ungroup() %>%
  left_join(ci) %>%
  left_join(model_14)

# Plot the entrance comparison.
plot_7 = data_16 %>%
  ggplot(aes(x=model, y=mean_human, label=map)) +
  geom_point(aes(color=substr(map, 0, 2))) +
  geom_smooth(method="lm", se=FALSE, linetype="dashed", color="grey") +
  ggtitle("Model Entrance Predictions vs. Participant Entrance Judgments") +
  xlab("Model Predictions") +
  ylab("Human Judgments") +
  ylim(0.0, 1.0) +
  scale_color_discrete(name="Entrance Type:") +
  theme_classic() +
  theme(aspect.ratio=1.0,
        plot.title=element_text(hjust=0.5),
        legend.title=element_text(hjust=0.5))
plot_7
```



For the entrance inferences under varying entrance priors, the Pearson correlation is r=0.85 (95% CI: 0.77-0.9). Next, we generate entrance inference comparisons for each trial.

```
# Plot entrance inferences by trial.
plot_8 = data_16 %>%
  mutate(map=gsub("_doors", "", map)) %>%
  gather(type, value, mean_human, model) %>%
  ggplot(aes(x=entrance, y=value, group=type)) +
  geom_point(aes(color=type)) +
  geom_line(aes(color=type)) +
  geom_errorbar(aes(ymin=lower, ymax=upper), color="#F8766D", width=0.2) +
  facet_wrap(~map, ncol=7) +
  xlab("Entrance") +
  ylab("Value") +
  scale_color_discrete(name="Observer Type:",
                       limits=c("mean_human", "model"),
                       labels=c("Human", "Model")) +
  theme_classic() +
  theme(aspect.ratio=1.0,
        legend.title=element_text(hjust=0.5),
        strip.text=element_text(size=5))
plot_8
```

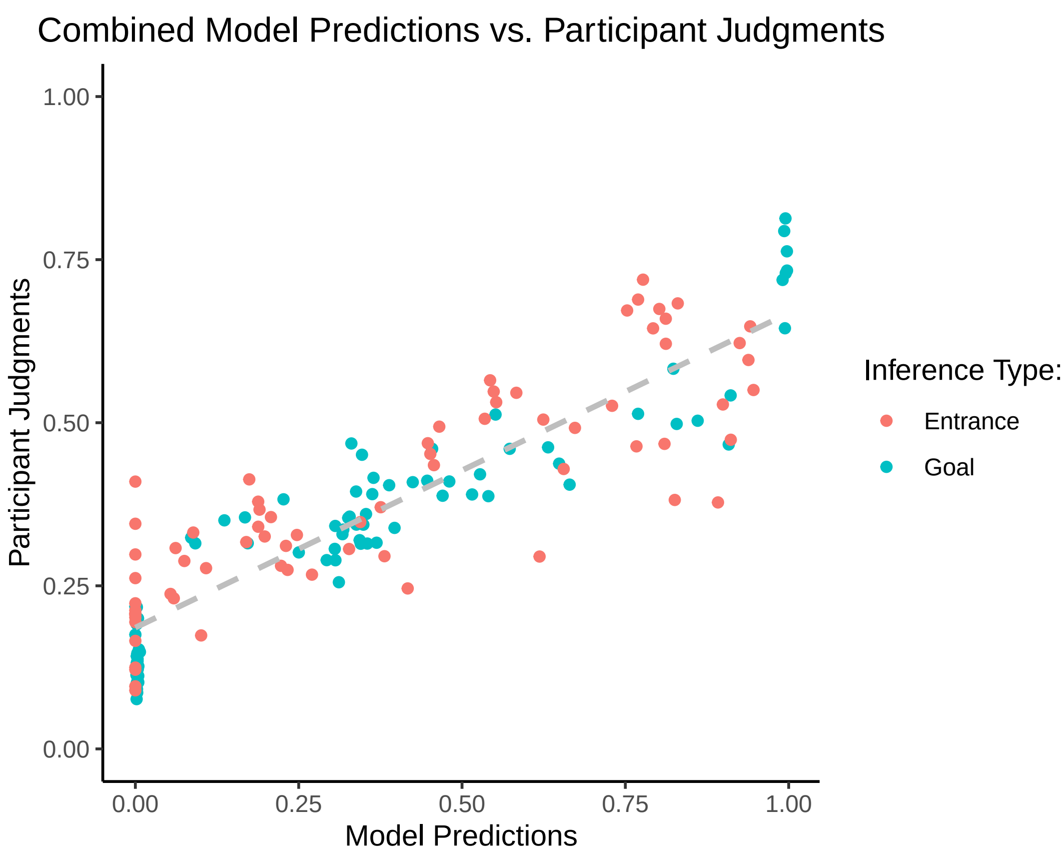


Combined Inference

Now we can combine both scatter plots.

```
# Merge the two inferences.
data_17 = data_13 %>%
  mutate(type="goal", inference=goal) %>%
  select(-goal) %>%
  rbind(select(mutate(data_16, type="entrance", inference=entrance), -entrance))

plot_9 = data_17 %>%
  ggplot(aes(x=model, y=mean_human, label=map)) +
  geom_point(aes(color=type)) +
  geom_smooth(method="lm", se=FALSE, linetype="dashed", color="grey") +
  ggtitle("Combined Model Predictions vs. Participant Judgments") +
  xlab("Model Predictions") +
  ylab("Participant Judgments") +
  ylim(0.0, 1.0) +
  scale_color_discrete(name="Inference Type:",
                       limits=c("entrance", "goal"),
                       labels=c("Entrance", "Goal")) +
  theme_classic() +
  theme(aspect.ratio=1.0,
        plot.title=element_text(hjust=0.5),
```

```
        legend.title=element_text(hjust=0.5))
plot_9
```



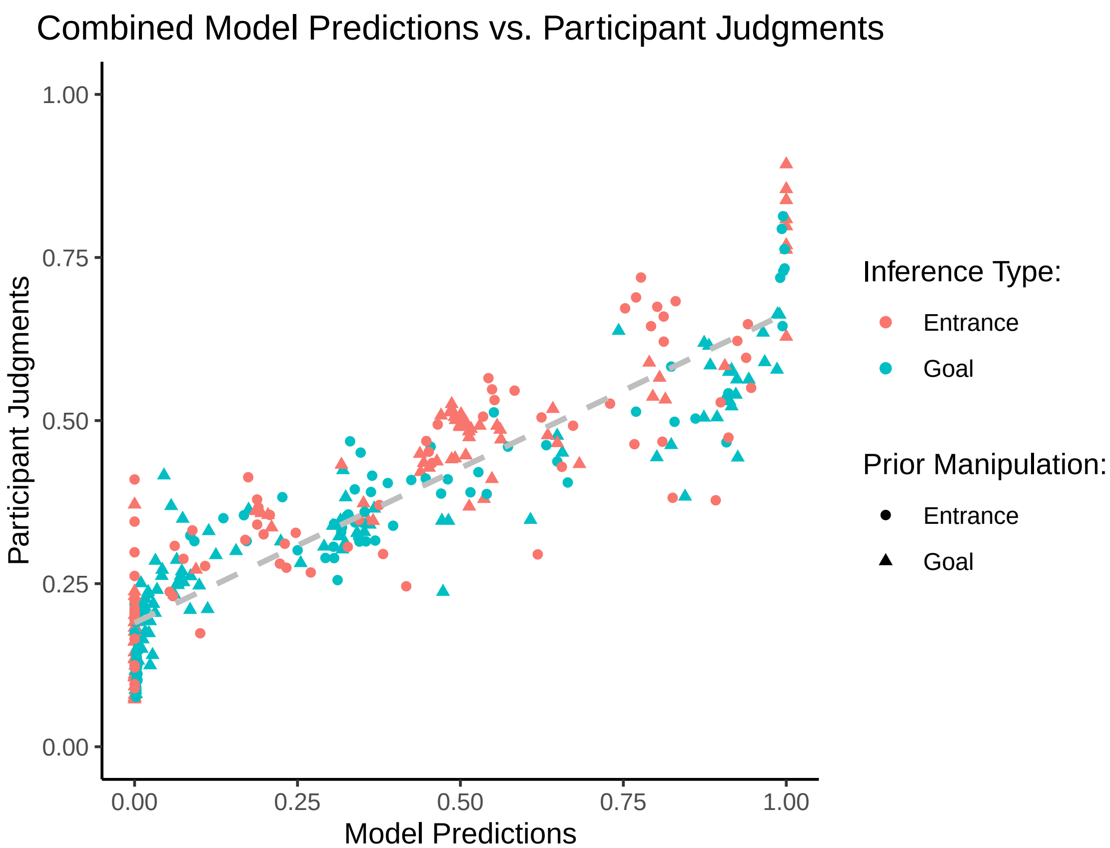
The combined Pearson correlation under varying entrance priors is r=0.9 (95% CI: 0.86-0.92).

Combining Prior Manipulations

Finally, we can combine the scatter plots from both prior manipulations.

```
# Merge the two inferences.
data_18 = data_10 %>%
  mutate(prior_manipulation="goal") %>%
  rbind(mutate(data_17, prior_manipulation="entrance"))

plot_10 = data_18 %>%
  ggplot(aes(x=model, y=mean_human, label=map)) +
  geom_point(aes(color=type, shape=prior_manipulation)) +
  geom_smooth(method="lm", se=FALSE, linetype="dashed", color="grey") +
  ggtitle("Combined Model Predictions vs. Participant Judgments") +
  xlab("Model Predictions") +
  ylab("Participant Judgments") +
  ylim(0.0, 1.0) +
  scale_color_discrete(name="Inference Type:",
```

```
                         limits=c("entrance", "goal"),
                         labels=c("Entrance", "Goal")) +
  scale_shape_discrete(name="Prior Manipulation:",
                         limits=c("entrance", "goal"),
                         labels=c("Entrance", "Goal")) +
  theme_classic() +
  theme(aspect.ratio=1.0,
        plot.title=element_text(hjust=0.5),
        legend.title=element_text(hjust=0.5))
plot_10
```



The combined Pearson correlation is r=0.91 (95% CI: 0.89-0.92).

Validating Prior Manipulations

Here we compare the data between Experiment 1 and Experiment 4 to highlight the effect that the prior manipulations had on the trials that were also used in Experiment 1. First, we need to set up the paths to the Experiment 1 data.

Next, we need to read in the goal inferences from Experiment 1.

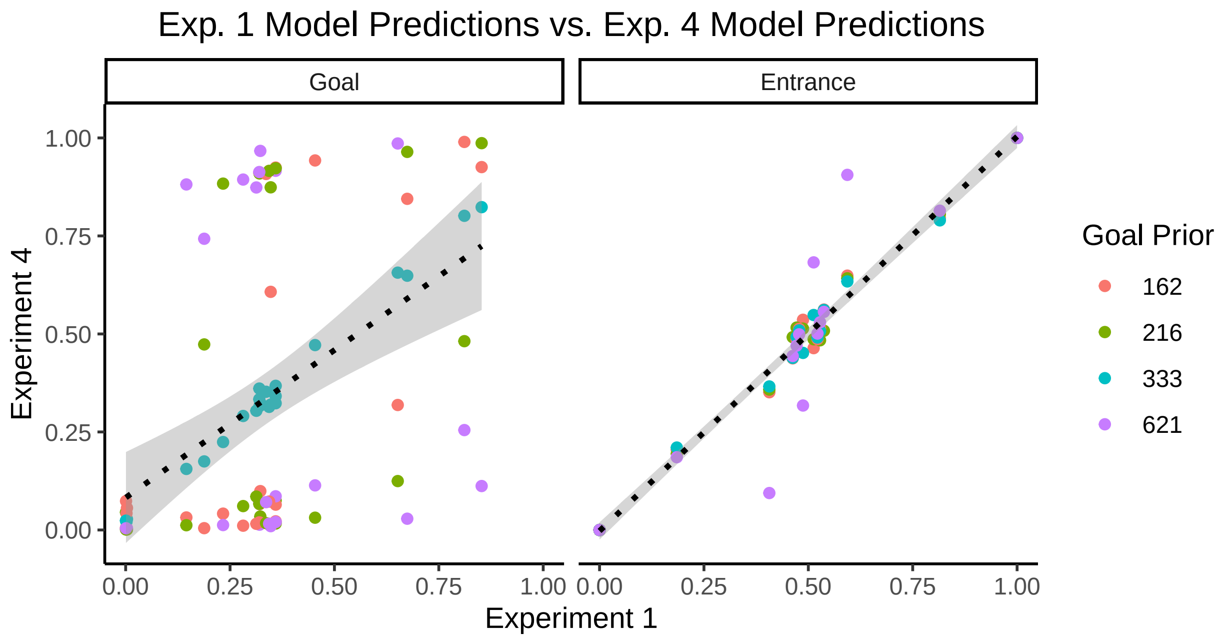
Then, we need to read in the entrance inferences from Experiment 1 and merge them with the goal inferences.

Goal Prior Manipulation

First, we'll compare the data between Experiment 1 and Experiment 4 where the goals were the manipulated prior information.

Model Predictions

```
# Split the map into the base map and the prior.
data_27 = data_10 %>%
  separate(map, into=c("map", "prior"), sep=7) %>%
  mutate(prior=gsub("_", "", prior))

# Filter the relevant trials from Experiment 1.
data_28 = data_26 %>%
  filter(map %in% (unique(data_27$map)))

# Generate the data frame comparing model predictions across experiments.
data_29 = data_27 %>%
  select(map, prior, model, z_model, type, inference) %>%
  rename(experiment_4=model, z_experiment_4=z_model) %>%
  left_join(select(data_28, map, model, z_model, type, inference) %>%
              rename(experiment_1=model, z_experiment_1=z_model))

# Plot the data.
plot_11 = data_29 %>%
  ggplot(aes(x=experiment_1, y=experiment_4, group=type)) +
  geom_point(aes(color=gsub("goals-", "", prior))) +
  geom_smooth(method="lm", se=TRUE, linetype="dotted", color="black") +
  ggtitle("Exp. 1 Model Predictions vs. Exp. 4 Model Predictions") +
  xlab("Experiment 1") +
  ylab("Experiment 4") +
  scale_color_discrete(name="Goal Prior") +
  facet_wrap(~factor(type,
                     levels=c("goal", "entrance"),
                     labels=c("Goal", "Entrance"))) +
  theme_classic() +
  theme(aspect.ratio=1.0,
        plot.title=element_text(hjust=0.5),
        legend.title=element_text(hjust=0.5))
plot_11
```



For the goal inferences under varying goal priors, the Pearson correlation is r=0.47 (95% CI: 0.3-0.61). For the entrance inferences under varying goal priors, the Pearson correlation is r=0.98 (95% CI: 0.95-0.95). In the table below, we compute the Pearson correlations of the goal and entrance inferences for each goal prior.

```
# Compute the correlation and its bootstrapped 95% CI (per inference type and
# prior).
set.seed(seed)
cor_8 = data_29 %>%
  group_by(type, prior) %>%
  summarize(cor=cor(experiment_1, experiment_4))
cor_8_ci = data.frame()
for (t in unique(data_29$type)) {
  for (p in unique(data_29$prior)) {
    # Filter the current inference type and prior.
    data_31 = data_29 %>%
      filter(type==t, prior==p)

    # Compute the bootstrap.
    cor_bootstrap = compute_bootstrap(data_31)

    # Store the bootstrapped 95% CIs for the current inference type and prior.
    cor_8_ci = rbind(cor_8_ci, data.frame(type=t,
                                          prior=p,
                                          lower=cor_bootstrap[4],
                                          upper=cor_bootstrap[5]))
  }
```

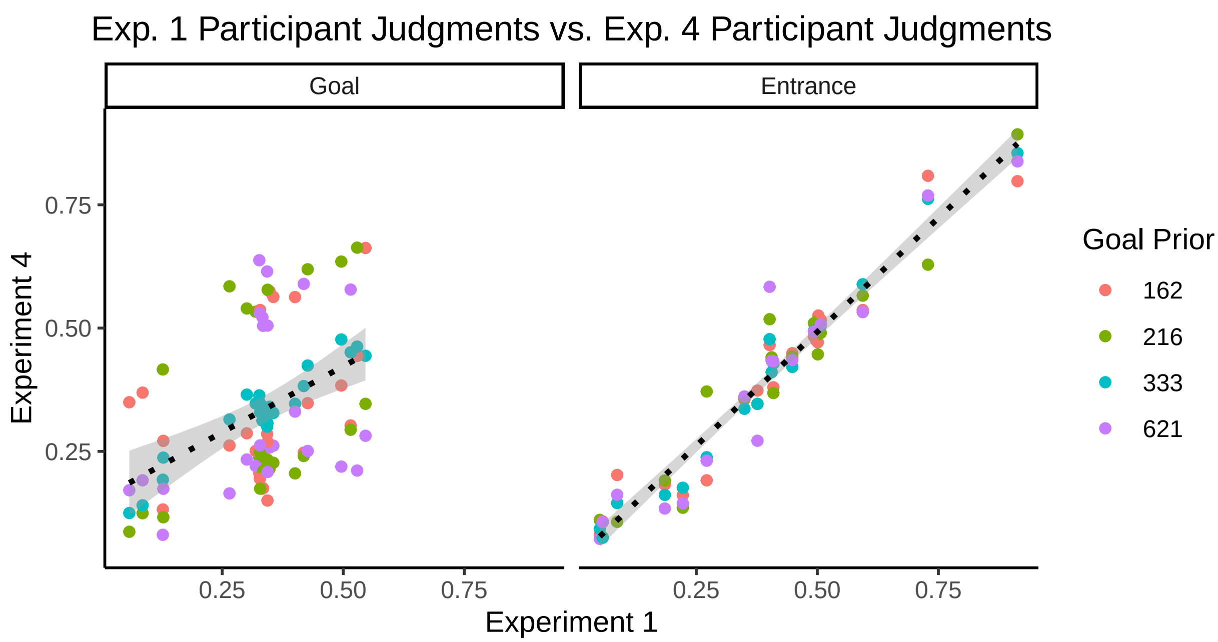
```
}

# Format and print the data frame.
table_0 = cor_8 %>%
  left_join(cor_8_ci) %>%
  mutate(type=factor(type,
                     levels=c("goal", "entrance"),
                     labels=c("Goal", "Entrance")),
         prior=gsub("goals-", "", prior),
         cor=round(cor, 2),
         lower=round(lower, 2),
         upper=round(upper, 2)) %>%
  kable(col.names=c("Inference Type", "Goal Prior", "$\\rho$",
                    "CI$_{95\\%}$ (lower)", "CI$_{95\\%}$ (upper)"))
table_0
```

Inference Type	Goal Prior	ρ	CI$_{95\%}$ (lower)	CI$_{95\%}$ (upper)
Entrance	162	1.00	0.99	1.00
Entrance	216	1.00	0.99	1.00
Entrance	333	1.00	0.99	1.00
Entrance	621	0.95	0.84	0.99
Goal	162	0.65	0.42	0.82
Goal	216	0.45	0.13	0.70
Goal	333	1.00	0.99	1.00
Goal	621	0.08	-0.26	0.43

Participant Judgments

```
# Generate the data frame comparing participant judgments across experiments.
data_32 = data_27 %>%
  select(map, prior, mean_human, mean_z_human, type, inference) %>%
  rename(experiment_4=mean_human, z_experiment_4=mean_z_human) %>%
  left_join(select(data_28, map, mean_human, mean_z_human, type, inference) %>%
              rename(experiment_1=mean_human, z_experiment_1=mean_z_human))

# Plot the data.
plot_12 = data_32 %>%
  ggplot(aes(x=experiment_1, y=experiment_4, group=type)) +
  geom_point(aes(color=gsub("goals-", "", prior))) +
  geom_smooth(method="lm", se=TRUE, linetype="dotted", color="black") +
  ggtitle("Exp. 1 Participant Judgments vs. Exp. 4 Participant Judgments") +
  xlab("Experiment 1") +
  ylab("Experiment 4") +
  scale_color_discrete(name="Goal Prior") +
  facet_wrap(~factor(type,
                     levels=c("goal", "entrance"),
                     labels=c("Goal", "Entrance"))) +
  theme_classic() +
  theme(aspect.ratio=1.0,
        plot.title=element_text(hjust=0.5),
```

```
       legend.title=element_text(hjust=0.5))
plot_12
```



For the goal inferences under varying goal priors, the Pearson correlation is r=0.45 (95% CI: 0.29-0.59). For the entrance inferences under varying goal priors, the Pearson correlation is r=0.97 (95% CI: 0.95-0.95). In the table below, we compute the Pearson correlations of the goal and entrance inferences for each goal prior.

```
# Compute the correlation and its bootstrapped 95% CI (per inference type and
# prior).
set.seed(seed)
cor_10 = data_32 %>%
  group_by(type, prior) %>%
  summarize(cor=cor(experiment_1, experiment_4))
cor_10_ci = data.frame()
for (t in unique(data_32$type)) {
  for (p in unique(data_32$prior)) {
    # Filter the current inference type and prior.
    data_34 = data_32 %>%
      filter(type==t, prior==p)

    # Compute the bootstrap.
    cor_bootstrap = compute_bootstrap(data_34)

    # Store the bootstrapped 95% CIs for the current inference type and prior.
    cor_10_ci = rbind(cor_10_ci, data.frame(type=t,
```

```
                                                prior=p,
                                                lower=cor_bootstrap[4],
                                                upper=cor_bootstrap[5]))
  }
}

# Format and print the data frame.
table_1 = cor_10 %>%
  left_join(cor_10_ci) %>%
  mutate(type=factor(type,
                     levels=c("goal", "entrance"),
                     labels=c("Goal", "Entrance")),
         prior=gsub("goals-", "", prior),
         cor=round(cor, 2),
         lower=round(lower, 2),
         upper=round(upper, 2)) %>%
  kable(col.names=c("Inference Type", "Goal Prior", "$\\rho$",
                    "CI$_{95\\%}$ (lower)", "CI$_{95\\%}$ (upper)"))
table_1
```

Inference Type	Goal Prior	ρ	$CI_{95\%}$ (lower)	$CI_{95\%}$ (upper)
Entrance	162	0.97	0.93	0.98
Entrance	216	0.97	0.91	0.99
Entrance	333	0.99	0.96	0.99
Entrance	621	0.96	0.84	0.98
Goal	162	0.35	0.06	0.64
Goal	216	0.43	0.04	0.70
Goal	333	0.95	0.89	0.98
Goal	621	0.35	-0.03	0.63

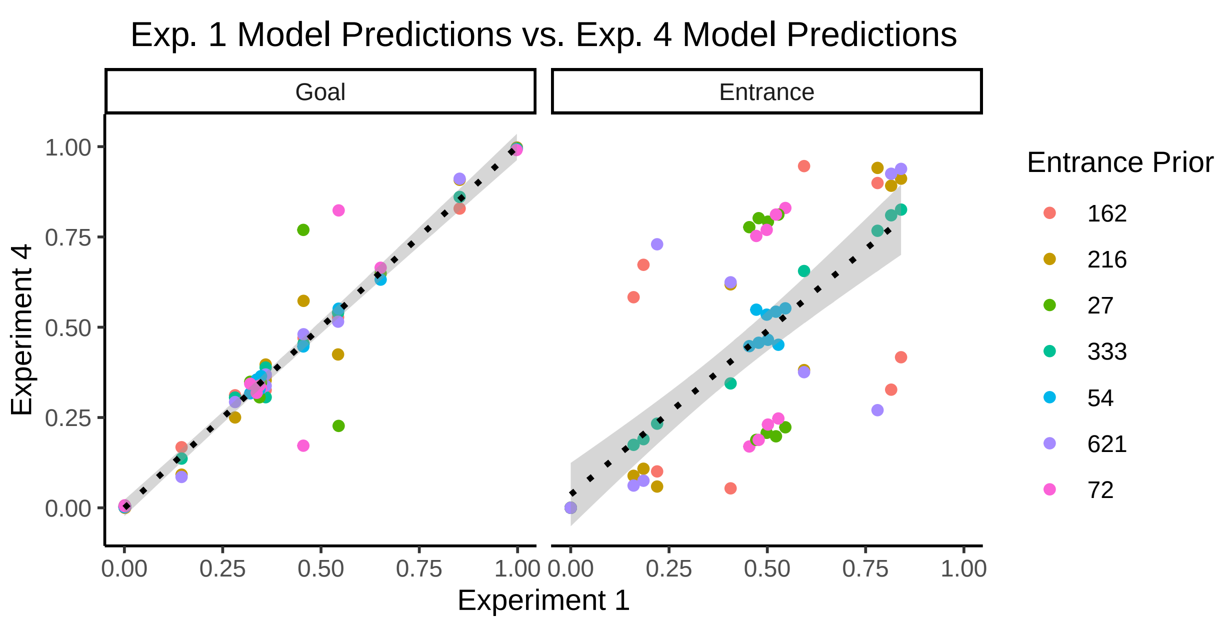
Entrance Prior Manipulation

Now, we'll compare the data between Experiment 1 and Experiment 4 where the entrances were the manipulated prior information.

Model Predictions

```
# Split the map into the base map and the prior.
data_35 = data_17 %>%
  separate(map, into=c("map", "prior"), sep=7) %>%
  mutate(prior=gsub("_", "", prior))

# Filter the relevant trials from Experiment 1.
data_36 = data_26 %>%
  filter(map %in% (unique(data_35$map)))

# Generate the data frame comparing model predictions across experiments.
data_37 = data_35 %>%
  select(map, prior, model, z_model, type, inference) %>%
```

```r
  rename(experiment_4=model, z_experiment_4=z_model) %>%
  left_join(select(data_36, map, model, z_model, type, inference) %>%
              rename(experiment_1=model, z_experiment_1=z_model))

# Plot the data.
plot_13 = data_37 %>%
  ggplot(aes(x=experiment_1, y=experiment_4, group=type)) +
  geom_point(aes(color=gsub("doors-", "", prior))) +
  geom_smooth(method="lm", se=TRUE, linetype="dotted", color="black") +
  ggtitle("Exp. 1 Model Predictions vs. Exp. 4 Model Predictions") +
  xlab("Experiment 1") +
  ylab("Experiment 4") +
  scale_color_discrete(name="Entrance Prior") +
  facet_wrap(~factor(type,
                     levels=c("goal", "entrance"),
                     labels=c("Goal", "Entrance"))) +
  theme_classic() +
  theme(aspect.ratio=1.0,
        plot.title=element_text(hjust=0.5),
        legend.title=element_text(hjust=0.5))
plot_13
```



For the goal inferences under varying goal priors, the Pearson correlation is r=0.98 (95% CI: 0.94-0.99). For the entrance inferences under varying goal priors, the Pearson correlation is r=0.76 (95% CI: 0.63-0.63). In the table below, we compute the Pearson correlations of the goal and entrance inferences for each goal prior.

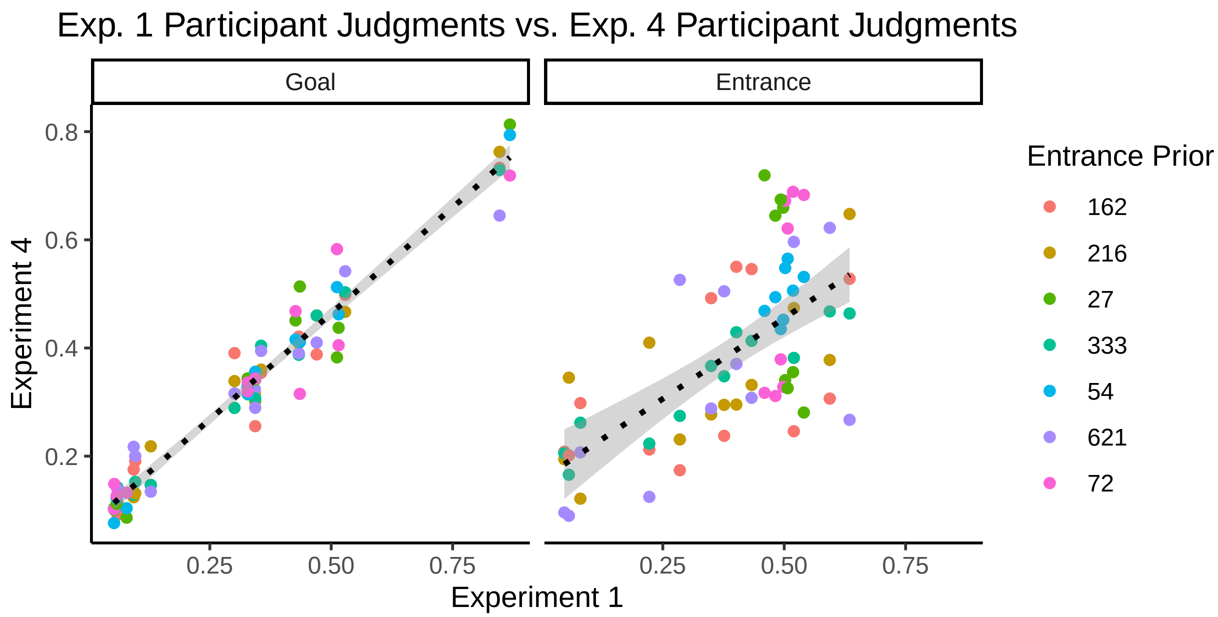
```r
# Compute the correlation and its bootstrapped 95% CI (per inference type and
# prior).
set.seed(seed)
cor_12 = data_37 %>%
  group_by(type, prior) %>%
  summarize(cor=cor(experiment_1, experiment_4))
cor_12_ci = data.frame()
for (t in unique(data_37$type)) {
  for (p in unique(data_37$prior)) {
    # Filter the current inference type and prior.
    data_39 = data_37 %>%
      filter(type==t, prior==p)

    # Compute the bootstrap.
    cor_bootstrap = compute_bootstrap(data_39)

    # Store the bootstrapped 95% CIs for the current inference type and prior.
    cor_12_ci = rbind(cor_12_ci, data.frame(type=t,
                                            prior=p,
                                            lower=cor_bootstrap[4],
                                            upper=cor_bootstrap[5]))
  }
}

# Format and print the data frame.
table_2 = cor_12 %>%
  left_join(cor_12_ci) %>%
  mutate(type=factor(type,
                     levels=c("goal", "entrance"),
                     labels=c("Goal", "Entrance")),
         prior=gsub("doors-", "", prior),
         cor=round(cor, 2),
         lower=round(lower, 2),
         upper=round(upper, 2)) %>%
  kable(col.names=c("Inference Type", "Entrance Prior", "$\\rho$",
                    "CI$_{95\\%}$ (lower)", "CI$_{95\\%}$ (upper)"))
table_2
```

Inference Type	Entrance Prior	ρ	$CI_{95\%}$ (lower)	$CI_{95\%}$ (upper)
Entrance	162	0.59	0.14	0.86
Entrance	216	0.96	0.82	0.99
Entrance	27	-0.29	-0.80	0.63
Entrance	333	1.00	0.98	1.00
Entrance	54	0.37	-0.58	0.89
Entrance	621	0.77	0.28	0.95
Entrance	72	0.43	-0.47	0.87
Goal	162	1.00	0.99	1.00
Goal	216	0.99	0.94	1.00
Goal	27	0.91	0.57	1.00
Goal	333	1.00	0.99	1.00
Goal	54	1.00	1.00	1.00
Goal	621	1.00	0.99	1.00

Inference Type	Entrance Prior	ρ	$CI_{95\%}$ (lower)	$CI_{95\%}$ (upper)
Goal	72	0.93	0.69	1.00

Participant Judgments

```
# Generate the data frame comparing participant judgments across experiments.
data_40 = data_35 %>%
  select(map, prior, mean_human, mean_z_human, type, inference) %>%
  rename(experiment_4=mean_human, z_experiment_4=mean_z_human) %>%
  left_join(select(data_36, map, mean_human, mean_z_human, type, inference) %>%
              rename(experiment_1=mean_human, z_experiment_1=mean_z_human))

# Plot the data.
plot_14 = data_40 %>%
  ggplot(aes(x=experiment_1, y=experiment_4, group=type)) +
  geom_point(aes(color=gsub("doors-", "", prior))) +
  geom_smooth(method="lm", se=TRUE, linetype="dotted", color="black") +
  ggtitle("Exp. 1 Participant Judgments vs. Exp. 4 Participant Judgments") +
  xlab("Experiment 1") +
  ylab("Experiment 4") +
  scale_color_discrete(name="Entrance Prior") +
  facet_wrap(~factor(type,
                     levels=c("goal", "entrance"),
                     labels=c("Goal", "Entrance"))) +
  theme_classic() +
  theme(aspect.ratio=1.0,
        plot.title=element_text(hjust=0.5),
        legend.title=element_text(hjust=0.5))
plot_14
```



For the goal inferences under varying goal priors, the Pearson correlation is r=0.98 (95% CI: 0.96-0.98). For the entrance inferences under varying goal priors, the Pearson correlation is r=0.65 (95% CI: 0.5-0.5). In the table below, we compute the Pearson correlations of the goal and entrance inferences for each goal prior.

```
# Compute the correlation and its bootstrapped 95% CI (per inference type and
# prior).
set.seed(seed)
cor_14 = data_40 %>%
  group_by(type, prior) %>%
  summarize(cor=cor(experiment_1, experiment_4))
cor_14_ci = data.frame()
for (t in unique(data_40$type)) {
  for (p in unique(data_40$prior)) {
    # Filter the current inference type and prior.
    data_42 = data_40 %>%
      filter(type==t, prior==p)

    # Compute the bootstrap.
    cor_bootstrap = compute_bootstrap(data_42)

    # Store the bootstrapped 95% CIs for the current inference type and prior.
    cor_14_ci = rbind(cor_14_ci, data.frame(type=t,
                                            prior=p,
                                            lower=cor_bootstrap[4],
                                            upper=cor_bootstrap[5]))
  }
```

```
}

# Format and print the data frame.
table_3 = cor_14 %>%
  left_join(cor_14_ci) %>%
  mutate(type=factor(type,
                     levels=c("goal", "entrance"),
                     labels=c("Goal", "Entrance")),
         prior=gsub("doors-", "", prior),
         cor=round(cor, 2),
         lower=round(lower, 2),
         upper=round(upper, 2)) %>%
  kable(col.names=c("Inference Type", "Entrance Prior", "$\\rho$",
                    "CI$_{95\\%}$ (lower)", "CI$_{95\\%}$ (upper)"))
table_3
```

Inference Type	Entrance Prior	ρ	$\mathrm{CI}_{95\%}$ (lower)	$\mathrm{CI}_{95\%}$ (upper)
Entrance	162	0.52	0.13	0.79
Entrance	216	0.70	0.15	0.92
Entrance	27	-0.82	-0.95	-0.56
Entrance	333	0.93	0.83	0.97
Entrance	54	0.50	0.07	0.79
Entrance	621	0.69	0.05	0.92
Entrance	72	0.80	0.40	0.91
Goal	162	0.97	0.86	0.99
Goal	216	0.99	0.96	1.00
Goal	27	0.97	0.89	0.99
Goal	333	0.99	0.96	1.00
Goal	54	1.00	0.99	1.00
Goal	621	0.96	0.91	0.99
Goal	72	0.96	0.86	0.99