Biased Experiment

```
library(conflicted)

library(kableExtra)
library(knitr)
library(broom.helpers)
library(broom)
library(dtplyr)
library(furrr)
```

```
## Loading required package: future
```

```
library(arrow)
library(glue)
library(fs)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
```

```
conflict_prefer("filter", "dplyr")
```

```
## [conflicted] Will prefer dplyr::filter over any other package.
```

```
source(here("analysis/utils.R"), local = knit_global())
set_theme()
```

```
write_bib(.packages(), here("analysis/packages.bib"))
sessionInfo()
```

```
## R version 4.4.0 (2024-04-24)
## Platform: aarch64-apple-darwin20
## Running under: macOS Sonoma 14.5
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib;  LAPACK v
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: Asia/Singapore
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
```

```
##
## other attached packages:
##  [1] lubridate_1.9.3      forcats_1.0.0        stringr_1.5.1
##  [4] dplyr_1.1.4          purrr_1.0.2          readr_2.1.5
##  [7] tidyr_1.3.1          tibble_3.2.1         ggplot2_3.5.1
## [10] tidyverse_2.0.0      fs_1.6.4             glue_1.7.0
## [13] arrow_16.1.0         furrr_0.3.1          future_1.33.2
## [16] dtplyr_1.3.1         broom_1.0.6          broom.helpers_1.15.0
## [19] knitr_1.47           kableExtra_1.4.0     conflicted_1.2.0
## [22] here_1.0.1
##
## loaded via a namespace (and not attached):
##  [1] gtable_0.3.5      xfun_0.45         tzdb_0.4.0        vctrs_0.6.5
##  [5] tools_4.4.0       generics_0.1.3    parallel_4.4.0    fansi_1.0.6
##  [9] pkgconfig_2.0.3   data.table_1.15.4 assertthat_0.2.1  lifecycle_1.0.4
## [13] compiler_4.4.0    munsell_0.5.1     codetools_0.2-20  htmltools_0.5.8.1
## [17] yaml_2.3.8        pillar_1.9.0      cachem_1.1.0      parallelly_1.37.1
## [21] tidyselect_1.2.1  digest_0.6.35     stringi_1.8.4     listenv_0.9.1
## [25] rprojroot_2.0.4   fastmap_1.2.0     grid_4.4.0        colorspace_2.1-0
## [29] cli_3.6.2         magrittr_2.0.3    utf8_1.2.4        withr_3.0.0
## [33] scales_1.3.0      backports_1.5.0   bit64_4.0.5       timechange_0.3.0
## [37] rmarkdown_2.27    globals_0.16.3    bit_4.0.5         hms_1.1.3
## [41] memoise_2.0.1     evaluate_0.24.0   viridisLite_0.4.2 rlang_1.1.4
## [45] xml2_1.3.6        svglite_2.1.3     rstudioapi_0.16.0 R6_2.5.1
## [49] systemfonts_1.1.0
```

Analyze attack trends

```
data_dir <- here(glue("{params$data}/{params$simulation}/results"))

success_fnames <-
  dir_ls(data_dir, glob = glue("*norm_{params$norm}*.csv"))

stopifnot(length(success_fnames) == 240)

# every fname is a simulation
success_raw_data <- get_data(success_fnames, read_csv) |>
  glimpse()
```

```
## Rows: 240
## Columns: 16
## $ fname                 <chr> "/Users/zbli/Documents/Documents - ZhaoBin's M~
## $ num_iteration         <dbl> 200, 200, 200, 200, 200, 200, 200, 200, 200, 2~
## $ max_norm              <dbl> 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05~
## $ model_name            <ord> Cascade R-CNN, Faster R-CNN, RetinaNet, SSD, Y~
## $ loss_target           <ord> Mislabeling, Mislabeling, Mislabeling, Mislabe~
## $ attack_bbox           <chr> "predictions", "predictions", "predictions", "~
## $ perturb_fun           <chr> "perturb_inside", "perturb_inside", "perturb_i~
## $ sample_count          <dbl> 1258, 1301, 703, 1105, 1157, 1258, 1301, 703, ~
## $ attack_count          <dbl> 100, 100, 100, 100, 100, 100, 100, 100, 100, 1~
## $ success_count         <dbl> 25, 23, 13, 62, 86, 47, 61, 58, 52, 51, 31, 25~
## $ vanish_count          <dbl> 15, 14, 5, 8, 39, 44, 59, 58, 48, 46, 30, 24, ~
## $ mislabel_count        <dbl> 10, 9, 8, 54, 47, 3, 2, 0, 4, 5, 1, 1, 1, 1, 2~
```

```
## $ mislabel_intended_count <dbl> 10, 9, 8, 54, 47, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ target_max_conf         <dbl> 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0~
## $ perturb_min_size        <dbl> 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25~
## $ bbox_max_dist           <dbl> 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25~
```

```
# target_max_conf, perturb_min_size, bbox_max_dist are the sampling criteria
success_raw_data <- success_raw_data |>
  rowwise() |>
  mutate(across(target_max_conf:bbox_max_dist, ~ !is.na(.)), # convert to TRUE/FALSE
    num_cri = sum(across(target_max_conf:bbox_max_dist))
  ) |>
  glimpse()
```

```
## Rows: 240
## Columns: 17
## Rowwise:
## $ fname                   <chr> "/Users/zbli/Documents/Documents - ZhaoBin's M~
## $ num_iteration           <dbl> 200, 200, 200, 200, 200, 200, 200, 200, 200, 2~
## $ max_norm                <dbl> 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05~
## $ model_name              <ord> Cascade R-CNN, Faster R-CNN, RetinaNet, SSD, Y~
## $ loss_target             <ord> Mislabeling, Mislabeling, Mislabeling, Mislabe~
## $ attack_bbox             <chr> "predictions", "predictions", "predictions", "~
## $ perturb_fun             <chr> "perturb_inside", "perturb_inside", "perturb_i~
## $ sample_count            <dbl> 1258, 1301, 703, 1105, 1157, 1258, 1301, 703, ~
## $ attack_count            <dbl> 100, 100, 100, 100, 100, 100, 100, 100, 100, 1~
## $ success_count           <dbl> 25, 23, 13, 62, 86, 47, 61, 58, 52, 51, 31, 25~
## $ vanish_count            <dbl> 15, 14, 5, 8, 39, 44, 59, 58, 48, 46, 30, 24, ~
## $ mislabel_count          <dbl> 10, 9, 8, 54, 47, 3, 2, 0, 4, 5, 1, 1, 1, 1, 2~
## $ mislabel_intended_count <dbl> 10, 9, 8, 54, 47, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ target_max_conf         <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE~
## $ perturb_min_size        <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE~
## $ bbox_max_dist           <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE~
## $ num_cri                 <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3~
```

```
# expand success per simulation into 1 and 0s per row
success_expanded_data <- success_raw_data |>
  rowwise() |>
  mutate(success = list(rep(0:1, times = c(attack_count - success_count, success_count)))) |>
  unnest_longer(success) |>
  glimpse()
```

```
## Rows: 24,000
## Columns: 18
## $ fname                   <chr> "/Users/zbli/Documents/Documents - ZhaoBin's M~
## $ num_iteration           <dbl> 200, 200, 200, 200, 200, 200, 200, 200, 200, 2~
## $ max_norm                <dbl> 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05~
## $ model_name              <ord> Cascade R-CNN, Cascade R-CNN, Cascade R-CNN, C~
## $ loss_target             <ord> Mislabeling, Mislabeling, Mislabeling, Mislabe~
## $ attack_bbox             <chr> "predictions", "predictions", "predictions", "~
## $ perturb_fun             <chr> "perturb_inside", "perturb_inside", "perturb_i~
## $ sample_count            <dbl> 1258, 1258, 1258, 1258, 1258, 1258, 1258, 1258~
## $ attack_count            <dbl> 100, 100, 100, 100, 100, 100, 100, 100, 100, 1~
## $ success_count           <dbl> 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25~
## $ vanish_count            <dbl> 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15~
## $ mislabel_count          <dbl> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10~
## $ mislabel_intended_count <dbl> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10~
```

```
## $ target_max_conf         <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE~
## $ perturb_min_size        <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE~
## $ bbox_max_dist           <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE~
## $ num_cri                 <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3~
## $ success                 <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

```
itr_lab <- "Number of Factors"

cap <- glue("{emp_tex('Success factors can be exploited in combination to significantly increase succes

cap
```

```
## Success factors can be exploited in combination to significantly increase success rates even with 0.
```

```
# use linear
g <- success_expanded_data |>
  ggplot(aes(num_cri, success, color = loss_target, linetype = loss_target)) +
  # use stat_summary rather than stat_summary_bin
  # since num_cri is set experimentally
  # mean_cl_boot gives 95% bootstrapped CI at 1000 samples
  # https://rdrr.io/cran/Hmisc/man/smean.sd.html
  stat_summary(fun.data = "mean_cl_boot") +
  binomial_smooth(formula = y ~ x) +
  facet_grid(cols = vars(model_name))

g +
  labs(x = itr_lab, y = glue("p(Success) {norm_axy(params$norm)}"), color = "Attack", linetype = "Attac
  scale_x_continuous(breaks = unique(success_raw_data$num_cri))
```

```
data <- success_expanded_data |>
  # avoid ordered regression
  mutate(
    model_name = factor(model_name, ordered = FALSE),
    loss_target = factor(loss_target, ordered = FALSE)
  ) |>
  glimpse()
```

```
## Rows: 24,000
## Columns: 18
## $ fname                   <chr> "/Users/zbli/Documents/Documents - ZhaoBin's M~
## $ num_iteration           <dbl> 200, 200, 200, 200, 200, 200, 200, 200, 200, 2~
## $ max_norm                <dbl> 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05~
## $ model_name              <fct> Cascade R-CNN, Cascade R-CNN, Cascade R-CNN, C~
## $ loss_target             <fct> Mislabeling, Mislabeling, Mislabeling, Mislabe~
## $ attack_bbox             <chr> "predictions", "predictions", "predictions", "~
## $ perturb_fun             <chr> "perturb_inside", "perturb_inside", "perturb_i~
## $ sample_count            <dbl> 1258, 1258, 1258, 1258, 1258, 1258, 1258, 1258~
## $ attack_count            <dbl> 100, 100, 100, 100, 100, 100, 100, 100, 100, 1~
## $ success_count           <dbl> 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25~
## $ vanish_count            <dbl> 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15~
## $ mislabel_count          <dbl> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10~
## $ mislabel_intended_count <dbl> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10~
## $ target_max_conf         <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE~
## $ perturb_min_size        <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE~
## $ bbox_max_dist           <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE~
## $ num_cri                 <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3~
```

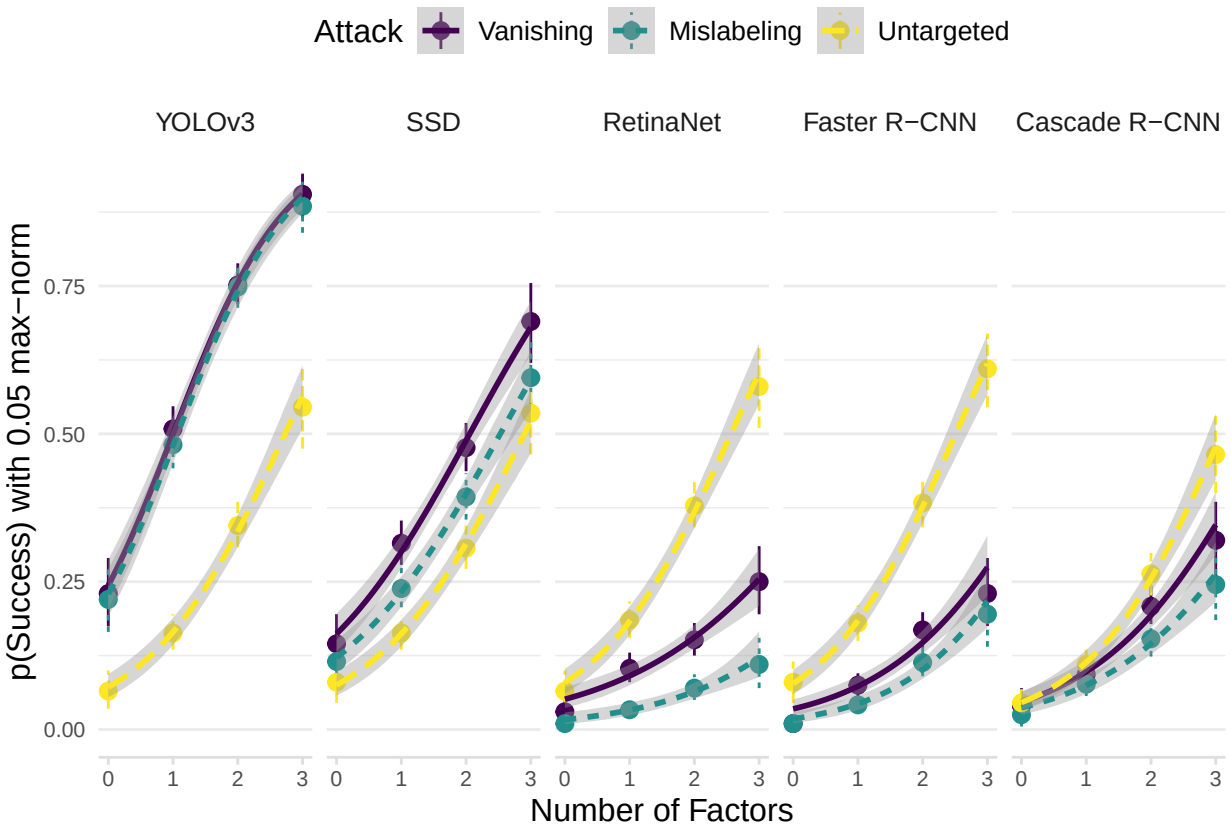

Figure 1: Success factors can be exploited in combination to significantly increase success rates even with 0.05 max-norm: We sampled target and perturb objects based on three validated success factors in Table **??** by targeting objects with low predicted confidence, perturbing large objects and selecting target and perturb objects close to one another. The binned summaries and regression trendlines graph success proportion against number of factors in the deliberate attack experiment. Errors are 95% confidence intervals and every point aggregates success over 200 images. Success rates significantly increase as the number of factors combined increases. Significance is determined at $\alpha < 0.05$ using a Wald z-test on the logistic estimates. Full details are given in Section **??**.

```
## $ success                 <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

```
model <- partial(glm_model, predictor = "num_cri")

reg_est <- get_tidied_reg(
  model, data
)
```

```
## `summarise()` has grouped output by 'model_name', 'loss_target'. You can
## override using the `.groups` argument.
```

```
ext_sig(reg_est, "pos")
```

```
## Total 15 predictors:
## 15 (100%) significant;
## 15 (100%) pos
```

```
## # A tibble: 15 x 9
## # Groups:   model_name, loss_target [15]
##    model_name    loss_target term  estimate std.error statistic p.value conf.low
```

```
##    <fct>         <fct>       <chr>    <dbl>     <dbl>     <dbl>   <dbl>    <dbl>
##  1 YOLOv3        Vanishing   num_~    1.13      0.075     15.1        0    0.987
##  2 YOLOv3        Mislabeling num_~    1.13      0.074     15.2        0    0.991
##  3 YOLOv3        Untargeted  num_~    0.938     0.076     12.3        0    0.791
##  4 SSD           Vanishing   num_~    0.8       0.067     12.0        0    0.671
##  5 SSD           Mislabeling num_~    0.781     0.069     11.3        0    0.647
##  6 SSD           Untargeted  num_~    0.864     0.076     11.4        0    0.718
##  7 RetinaNet     Vanishing   num_~    0.615     0.091      6.76       0    0.439
##  8 RetinaNet     Mislabeling num_~    0.703     0.137      5.13       0    0.438
##  9 RetinaNet     Untargeted  num_~    0.956     0.075     12.8        0    0.812
## 10 Faster R-CNN  Vanishing   num_~    0.783     0.097      8.06       0    0.595
## 11 Faster R-CNN  Mislabeling num_~    0.903     0.117      7.73       0    0.678
## 12 Faster R-CNN  Untargeted  num_~    0.985     0.075     13.1        0    0.84
## 13 Cascade R-CNN Vanishing   num_~    0.794     0.088      9.07       0    0.625
## 14 Cascade R-CNN Mislabeling num_~    0.743     0.097      7.69       0    0.556
## 15 Cascade R-CNN Untargeted  num_~    0.982     0.084     11.7        0    0.82
## # i 1 more variable: conf.high <dbl>
```

```
cap <- table_caption(glue("log({itr_lab})"), "Success rates increase with the number of factors combine

print_statistics(reg_est, cap)
```

Table 1: We run a logistic model regressing success against log(number of factors) in the randomized attack experiment. Success rates increase with the number of factors combined to select target and perturb objects for all models and attacks. Table headers are explained in Appendix **??**.

Group	Regression							
Attack	term	sig	estimate	std.error	statistic	p.value	conf.low	conf.high
YOLOv3								
Vanishing	num_cri	*	1.132	0.075	15.133	0	0.987	1.281
Mislabeling	num_cri	*	1.134	0.074	15.240	0	0.991	1.283
Untargeted	num_cri	*	0.938	0.076	12.307	0	0.791	1.090
SSD								
Vanishing	num_cri	*	0.800	0.067	11.974	0	0.671	0.933
Mislabeling	num_cri	*	0.781	0.069	11.291	0	0.647	0.919
Untargeted	num_cri	*	0.864	0.076	11.408	0	0.718	1.015
RetinaNet								
Vanishing	num_cri	*	0.615	0.091	6.760	0	0.439	0.795
Mislabeling	num_cri	*	0.703	0.137	5.131	0	0.438	0.976
Untargeted	num_cri	*	0.956	0.075	12.780	0	0.812	1.105
Faster R-CNN								
Vanishing	num_cri	*	0.783	0.097	8.062	0	0.595	0.976
Mislabeling	num_cri	*	0.903	0.117	7.729	0	0.678	1.136
Untargeted	num_cri	*	0.985	0.075	13.115	0	0.840	1.135
Cascade R-CNN								
Vanishing	num_cri	*	0.794	0.088	9.069	0	0.625	0.968
Mislabeling	num_cri	*	0.743	0.097	7.689	0	0.556	0.936
Untargeted	num_cri	*	0.982	0.084	11.703	0	0.820	1.149

```r
success_expanded_data |>
  group_by(model_name, loss_target, num_cri) |>
  summarize(mean(success))
```

```
## `summarise()` has grouped output by 'model_name', 'loss_target'. You can
## override using the `.groups` argument.
```

```
## # A tibble: 60 x 4
## # Groups:   model_name, loss_target [15]
##    model_name loss_target num_cri `mean(success)`
##    <ord>      <ord>         <int>           <dbl>
##  1 YOLOv3     Vanishing         0           0.23
##  2 YOLOv3     Vanishing         1           0.508
##  3 YOLOv3     Vanishing         2           0.752
##  4 YOLOv3     Vanishing         3           0.905
##  5 YOLOv3     Mislabeling       0           0.22
##  6 YOLOv3     Mislabeling       1           0.482
##  7 YOLOv3     Mislabeling       2           0.748
##  8 YOLOv3     Mislabeling       3           0.885
##  9 YOLOv3     Untargeted        0           0.065
## 10 YOLOv3     Untargeted        1           0.163
## # i 50 more rows
```