Arbitrary 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) == 960)

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

```
## Rows: 960
## 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              <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ 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> 51, 50, 50, 51, 50, 51, 50, 50, 51, 50, 51, 50~
## $ attack_count          <dbl> 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50~
## $ success_count         <dbl> 12, 12, 6, 22, 30, 13, 16, 10, 20, 17, 18, 12,~
## $ vanish_count          <dbl> 1, 0, 2, 6, 8, 11, 14, 6, 13, 13, 17, 11, 15, ~
## $ mislabel_count        <dbl> 11, 12, 4, 16, 22, 2, 2, 4, 7, 4, 1, 1, 1, 2, ~
```

```
## $ mislabel_intended_count <dbl> 11, 12, 4, 16, 22, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ target_max_conf         <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ perturb_min_size        <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ bbox_max_dist           <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ bbox_length             <dbl> 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0~
## $ boundary_distance       <dbl> 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01~
```

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

```
## Rows: 48,000
## Columns: 19
## $ fname                   <chr> "/Users/zbli/Documents/Documents - ZhaoBin's M~
## $ num_iteration           <dbl> 200, 200, 200, 200, 200, 200, 200, 200, 200, 2~
## $ max_norm                <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ 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> 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51~
## $ attack_count            <dbl> 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50~
## $ success_count           <dbl> 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12~
## $ vanish_count            <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ mislabel_count          <dbl> 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11~
## $ mislabel_intended_count <dbl> 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11~
## $ target_max_conf         <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ perturb_min_size        <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ bbox_max_dist           <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ bbox_len                <dbl> 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0~
## $ bbox_dist               <dbl> 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01~
## $ success                 <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

```
# control both
model <- partial(glm_model, predictor = "bbox_dist * bbox_len")
data <- success_expanded_data

reg_res <- get_tidied_reg(model, data, return_mod = TRUE)
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

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

```r
reg_est <- reg_res$tidied

ext_sig(reg_est, "neg", "bbox_dist")
```

```
## ----------bbox_dist----------
## Total 15 predictors:
## 15 (100%) significant;
## 15 (100%) neg
```

```
## # A tibble: 15 x 9
## # Groups:   model_name, loss_target [15]
##    model_name    loss_target term  estimate std.error statistic p.value conf.low
##    <ord>         <ord>       <chr>    <dbl>     <dbl>     <dbl>   <dbl>    <dbl>
##  1 YOLOv3        Vanishing   bbox~    -7.15      1.24     -5.75       0    -9.61
##  2 YOLOv3        Mislabeling bbox~    -7.54      1.24     -6.09       0    -9.99
##  3 YOLOv3        Untargeted  bbox~    -9.46      1.47     -6.44       0   -12.4
##  4 SSD           Vanishing   bbox~    -9.99      1.27     -7.88       0   -12.5
##  5 SSD           Mislabeling bbox~   -10.6       1.35     -7.83       0   -13.3
##  6 SSD           Untargeted  bbox~   -10.8       1.41     -7.65       0   -13.6
##  7 RetinaNet     Vanishing   bbox~   -17.7       2.72     -6.50       0   -23.2
##  8 RetinaNet     Mislabeling bbox~   -14.1       3.52     -4.02       0   -21.4
##  9 RetinaNet     Untargeted  bbox~   -16.0       2.00     -7.96       0   -20.0
## 10 Faster R-CNN  Vanishing   bbox~   -19.5       3.18     -6.15       0   -26.0
## 11 Faster R-CNN  Mislabeling bbox~   -19.0       3.68     -5.15       0   -26.5
## 12 Faster R-CNN  Untargeted  bbox~   -19.5       2.00     -9.72       0   -23.5
## 13 Cascade R-CNN Vanishing   bbox~   -24.8       3.45     -7.19       0   -31.8
## 14 Cascade R-CNN Mislabeling bbox~   -28.5       4.59     -6.21       0   -37.9
## 15 Cascade R-CNN Untargeted  bbox~   -34.5       3.09    -11.2        0   -40.7
## # i 1 more variable: conf.high <dbl>
```

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

```
## ----------bbox_len----------
## 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
##    <ord>         <ord>       <chr>    <dbl>     <dbl>     <dbl>   <dbl>    <dbl>
##  1 YOLOv3        Vanishing   bbox~     7.65     0.578      13.2       0     6.54
##  2 YOLOv3        Mislabeling bbox~     6.06     0.442      13.7       0     5.20
##  3 YOLOv3        Untargeted  bbox~     2.90     0.287      10.1       0     2.34
##  4 SSD           Vanishing   bbox~     4.19     0.326      12.8       0     3.56
##  5 SSD           Mislabeling bbox~     5.54     0.362      15.3       0     4.84
##  6 SSD           Untargeted  bbox~     3.50     0.296      11.8       0     2.92
##  7 RetinaNet     Vanishing   bbox~     3.48     0.353       9.85      0     2.79
##  8 RetinaNet     Mislabeling bbox~     2.44     0.399       6.13      0     1.66
##  9 RetinaNet     Untargeted  bbox~     3.48     0.327      10.7       0     2.85
## 10 Faster R-CNN  Vanishing   bbox~     3.24     0.36        8.99      0     2.54
## 11 Faster R-CNN  Mislabeling bbox~     2.00     0.386       5.19      0     1.25
## 12 Faster R-CNN  Untargeted  bbox~     3.01     0.31        9.69      0     2.40
## 13 Cascade R-CNN Vanishing   bbox~     4.50     0.41       11.0       0     3.70
## 14 Cascade R-CNN Mislabeling bbox~     3.12     0.391       7.98      0     2.36
```

```
## 15 Cascade R-CNN Untargeted  bbox~     1.75     0.314      5.56       0     1.13
## # i 1 more variable: conf.high <dbl>
```

```
ext_sig(reg_est, "both", "bbox_dist:bbox_len")
```

```
## ----------bbox_dist:bbox_len----------
## Total 15 predictors:
## 10 (67%) significant;
## 10 (67%) both
```

```
## # A tibble: 10 x 9
## # Groups:   model_name, loss_target [10]
##    model_name    loss_target term  estimate std.error statistic p.value conf.low
##    <ord>         <ord>       <chr>    <dbl>     <dbl>     <dbl>   <dbl>    <dbl>
##  1 YOLOv3        Vanishing   bbox~   -12.2       3.88     -3.16   0.002    -19.9
##  2 SSD           Mislabeling bbox~    -7.15      2.98     -2.40   0.016    -13.0
##  3 RetinaNet     Vanishing   bbox~   -27.2       6.14     -4.44   0        -39.3
##  4 RetinaNet     Mislabeling bbox~   -23.9       7.83     -3.06   0.002    -39.2
##  5 RetinaNet     Untargeted  bbox~    24.4       3.64      6.69   0         17.3
##  6 Faster R-CNN  Vanishing   bbox~   -24.0       6.89     -3.49   0        -37.5
##  7 Faster R-CNN  Untargeted  bbox~    26.4       3.61      7.32   0         19.4
##  8 Cascade R-CNN Vanishing   bbox~   -38.8       7.93     -4.89   0        -54.3
##  9 Cascade R-CNN Mislabeling bbox~   -20.4       9.40     -2.17   0.03     -38.7
## 10 Cascade R-CNN Untargeted  bbox~    39.2       5.00      7.83   0         29.5
## # i 1 more variable: conf.high <dbl>
```

```
dist_lab <- "Perturb-Target Distance"
len_lab <- "Perturb Box Length"

pred_name <- glue("{dist_lab} and {len_lab}, both relative to image width or height,")
main_pt <- glue("longer {len_lab} or shorter {dist_lab} cause success rates to significantly increase f

print_statistics(reg_est, table_caption(pred_name, main_pt, "deliberate"))
```

Table 1: We run a logistic model regressing success against perturb-target distance and perturb box length, both relative to image width or height, in the deliberate attack experiment. Longer perturb box length or shorter perturb-target distance cause success rates to significantly increase for all model and attack combinations, except for perturb box length in untargeted attack on Cascade R-CNN. The interaction terms, even when significant, are negligibly close to 0. Table headers are explained in Appendix **??**.

Group		Regression						
Attack	term	sig	estimate	std.error	statistic	p.value	conf.low	conf.high
YOLOv3								
Vanishing	distance	*	-7.152	1.243	-5.753	0.000	-9.610	-4.734
	length	*	7.648	0.578	13.235	0.000	6.543	8.810
	distance * length	*	-12.247	3.877	-3.159	0.002	-19.885	-4.676
Mislabeling	distance	*	-7.541	1.239	-6.087	0.000	-9.993	-5.135
	length	*	6.055	0.442	13.713	0.000	5.205	6.937
	distance * length		0.465	3.465	0.134	0.893	-6.299	7.295
Untargeted	distance	*	-9.464	1.469	-6.441	0.000	-12.392	-6.629
	length	*	2.895	0.287	10.081	0.000	2.336	3.463

	distance * length		4.370	2.862	1.527	0.127	-1.201	10.021
SSD								
Vanishing	distance	*	-9.986	1.267	-7.881	0.000	-12.501	-7.532
	length	*	4.189	0.326	12.840	0.000	3.556	4.835
	distance * length		-1.319	2.772	-0.476	0.634	-6.734	4.138
Mislabeling	distance	*	-10.593	1.354	-7.826	0.000	-13.284	-7.975
	length	*	5.541	0.362	15.323	0.000	4.841	6.259
	distance * length	*	-7.154	2.976	-2.404	0.016	-12.974	-1.302
Untargeted	distance	*	-10.787	1.410	-7.652	0.000	-13.594	-8.065
	length	*	3.497	0.296	11.810	0.000	2.921	4.082
	distance * length		1.528	2.835	0.539	0.590	-3.998	7.119
RetinaNet								
Vanishing	distance	*	-17.682	2.722	-6.496	0.000	-23.208	-12.539
	length	*	3.479	0.353	9.849	0.000	2.793	4.178
	distance * length	*	-27.250	6.138	-4.440	0.000	-39.253	-15.183
Mislabeling	distance	*	-14.139	3.516	-4.022	0.000	-21.420	-7.626
	length	*	2.442	0.399	6.127	0.000	1.665	3.227
	distance * length	*	-23.945	7.834	-3.056	0.002	-39.181	-8.436
Untargeted	distance	*	-15.950	2.003	-7.964	0.000	-19.953	-12.100
	length	*	3.483	0.327	10.664	0.000	2.850	4.130
	distance * length	*	24.373	3.645	6.687	0.000	17.330	31.623
Faster R-CNN								
Vanishing	distance	*	-19.538	3.179	-6.146	0.000	-26.021	-13.562
	length	*	3.241	0.360	8.995	0.000	2.541	3.953
	distance * length	*	-24.042	6.889	-3.490	0.000	-37.462	-10.448
Mislabeling	distance	*	-18.953	3.679	-5.151	0.000	-26.533	-12.110
	length	*	2.001	0.386	5.187	0.000	1.249	2.762
	distance * length		-14.029	7.793	-1.800	0.072	-29.166	1.402
Untargeted	distance	*	-19.478	2.004	-9.722	0.000	-23.486	-15.630
	length	*	3.007	0.310	9.694	0.000	2.404	3.620
	distance * length	*	26.412	3.607	7.322	0.000	19.439	33.585
Cascade R-CNN								
Vanishing	distance	*	-24.815	3.450	-7.193	0.000	-31.799	-18.282
	length	*	4.498	0.410	10.967	0.000	3.704	5.312
	distance * length	*	-38.766	7.932	-4.887	0.000	-54.349	-23.234
Mislabeling	distance	*	-28.520	4.590	-6.214	0.000	-37.922	-19.941
	length	*	3.122	0.391	7.978	0.000	2.362	3.896
	distance * length	*	-20.448	9.401	-2.175	0.030	-38.672	-1.816
Untargeted	distance	*	-34.458	3.088	-11.159	0.000	-40.684	-28.577
	length	*	1.746	0.314	5.556	0.000	1.134	2.367

distance * length	*	39.168	5.001	7.832	0.000	29.539	49.150

```
reg_mod <- reg_res$mod

newdata <- expand_grid(
  bbox_dist = linear_space(data$bbox_dist),
  bbox_len = unique(data$bbox_len)
) |>
  glimpse()
```

```
## Rows: 400
## Columns: 2
## $ bbox_dist <dbl> 0.01000000, 0.01000000, 0.01000000, 0.01000000, 0.01191919, ~
## $ bbox_len  <dbl> 0.1, 0.3, 0.5, 0.7, 0.1, 0.3, 0.5, 0.7, 0.1, 0.3, 0.5, 0.7, ~
```

```
# type.predict = "link" by default
# https://broom.tidymodels.org/reference/augment.glm.html
# https://stackoverflow.com/questions/14423325/confidence-intervals-for-predictions-from-logistic-regre
reg_pred <- reg_mod |>
  summarize(augment(mod, newdata = newdata, se_fit = TRUE)) |>
  mutate(success = plogis(.fitted), ul = plogis(.fitted + 1.96 * .se.fit), ll = plogis(.fitted - 1.96 *
  glimpse()
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

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

```
## Rows: 6,000
## Columns: 9
## Groups: model_name, loss_target [15]
## $ model_name  <ord> YOLOv3, YOLOv3, YOLOv3, YOLOv3, YOLOv3, YOLOv3, YOLOv3, YO~
## $ loss_target <ord> Vanishing, Vanishing, Vanishing, Vanishing, Vanishing, Van~
## $ bbox_dist   <dbl> 0.01000000, 0.01000000, 0.01000000, 0.01000000, 0.01191919~
## $ bbox_len    <dbl> 0.1, 0.3, 0.5, 0.7, 0.1, 0.3, 0.5, 0.7, 0.1, 0.3, 0.5, 0.7~
## $ .fitted     <dbl> 0.6353413, 2.1403805, 3.6454197, 5.1504590, 0.6192650, 2.1~
## $ .se.fit     <dbl> 0.10429738, 0.09666369, 0.17756249, 0.27826543, 0.10290874~
## $ success     <dbl> 0.6536996, 0.8947664, 0.9745540, 0.9942367, 0.6500514, 0.8~
## $ ul          <dbl> 0.6984155, 0.9113185, 0.9818976, 0.9966514, 0.6944414, 0.9~
## $ ll          <dbl> 0.6060930, 0.8755469, 0.9643394, 0.9900979, 0.6029002, 0.8~
```

```
arb_cap <- glue("{norm_tex('Perturbing an arbitrary region obfuscates intent with increased success for
```

```
## Warning in norm_tex("Perturbing an arbitrary region obfuscates intent with
## increased success for all models and attacks", : NAs introduced by coercion
```

```
arb_cap
```

```
## Perturbing an arbitrary region obfuscates intent with increased success for all models and attacks W
```

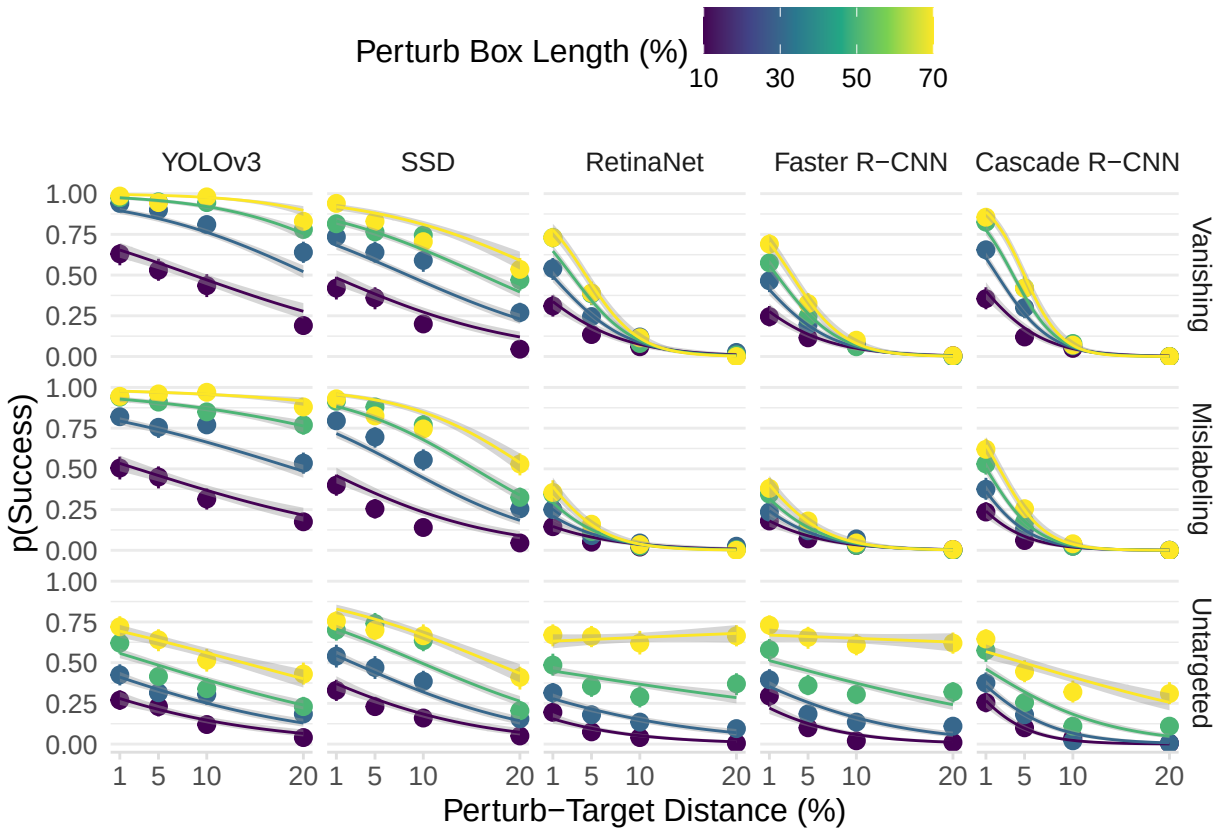

Figure 1: Perturbing an arbitrary region obfuscates intent with increased success for all models and attacks We implement intent obfuscating attack by perturbing an arbitrary non-overlapping square region to disrupt a randomly selected target object at various lengths and distances. The binned summaries and regression trendlines graph success proportion against perturb-target distance and perturb box length, both relative to image width or height, in the deliberate attack experiment. Errors are 95% confidence intervals and every point aggregates success over 200 images. The deliberate attack multiplies success as compared to the randomized attack (Figure **??**), especially at close perturb-target distance and large perturb box length. Full details are given in Section **??**.

```
g <- success_expanded_data |> ggplot(aes(bbox_dist, success, color = bbox_len, group = bbox_len)) +
  stat_summary(fun.data = "mean_cl_boot") +
  facet_grid(cols = vars(model_name), rows = vars(loss_target))

# https://github.com/tidyverse/ggplot2/blob/ef00be7e2016e1259b4aef7f7c85651df123beff/R/geom-smooth.r#L1
g <- g + geom_ribbon(
  data = reg_pred, aes(ymin = ll, ymax = ul),
  fill = "grey60", linetype = 0, alpha = 0.4
) +
  geom_line(data = reg_pred)

g + labs(x = glue("{dist_lab} (%)"), y = glue("p(Success) {norm_axy(params$norm)}")) +
  scale_x_continuous(breaks = unique(success_expanded_data$bbox_dist), labels = scales::label_percent(su
  scale_color_viridis_c(name = glue("{len_lab} (%)"), breaks = unique(success_expanded_data$bbox_len),
```

```
## Warning in norm_axy(params$norm): NAs introduced by coercion
```

```r
get_reg_vars <- function(data) {
  data |> select(bbox_dist, bbox_size_perturb, model_name, loss_target, success, object)
}

# run random.Rmd 1st
rand_dist_size <- readRDS(here("analysis/rand_dist_size.RDS")) |>
  mutate(object = 1) |>
  get_reg_vars() |>
  glimpse()
```

```
## Rows: 60,000
## Columns: 6
## $ bbox_dist         <dbl> 0.48728447, 0.38997352, 0.16133960, 0.01849709, 0.46~
## $ bbox_size_perturb <dbl> 0.0017605700, 0.0020902666, 0.0392784101, 0.07321143~
## $ model_name        <ord> Cascade R-CNN, Cascade R-CNN, Cascade R-CNN, Cascade~
## $ loss_target       <ord> Mislabeling, Mislabeling, Mislabeling, Mislabeling, ~
## $ success           <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ object            <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
```

```r
comb_dist_size <- success_expanded_data |>
  mutate(object = 0, bbox_size_perturb = bbox_len^2) |>
  get_reg_vars() |>
  bind_rows(rand_dist_size) |>
  mutate(
    bbox_dist = bbox_dist,
    bbox_size_perturb = bbox_size_perturb
  ) |>
  glimpse()
```

```
## Rows: 108,000
## Columns: 6
## $ bbox_dist         <dbl> 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01~
## $ bbox_size_perturb <dbl> 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01~
## $ model_name        <ord> Cascade R-CNN, Cascade R-CNN, Cascade R-CNN, Cascade~
## $ loss_target       <ord> Mislabeling, Mislabeling, Mislabeling, Mislabeling, ~
## $ success           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ object            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

```r
stopifnot(nrow(comb_dist_size) == nrow(success_expanded_data) +
  nrow(rand_dist_size) && sum(is.na(comb_dist_size)) == 0)
```

```r
# control both
model <- partial(glm_model, predictor = "object + bbox_dist * bbox_size_perturb")
data <- comb_dist_size

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

```
## Warning: There were 2 warnings in `mutate()`.
## The first warning was:
## i In argument: `mod = list(model(data))`.
## i In row 13.
## Caused by warning:
## ! glm.fit: fitted probabilities numerically 0 or 1 occurred
## i Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.
```

```
## Warning: There were 128 warnings in `summarize()`.
```

```
## The first warning was:
## i In argument: `tidy_plus_plus(mod, conf.int = TRUE)`.
## i In row 7.
## Caused by warning:
## ! glm.fit: fitted probabilities numerically 0 or 1 occurred
## i Run `dplyr::last_dplyr_warnings()` to see the 127 remaining warnings.
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

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

```
ext_sig(reg_est, "neg", "object")
```

```
## ----------object----------
## Total 15 predictors:
## 12 (80%) significant;
## 11 (73%) neg
```

```
## # A tibble: 11 x 9
## # Groups:   model_name, loss_target [11]
##    model_name    loss_target term  estimate std.error statistic p.value conf.low
##    <ord>         <ord>       <chr>    <dbl>     <dbl>     <dbl>   <dbl>    <dbl>
##  1 YOLOv3        Vanishing   obje~   -0.537     0.069     -7.79   0       -0.673
##  2 YOLOv3        Mislabeling obje~   -0.622     0.064     -9.73   0       -0.747
##  3 YOLOv3        Untargeted  obje~   -0.776     0.077    -10.1    0       -0.928
##  4 RetinaNet     Vanishing   obje~   -0.251     0.085     -2.95   0.003   -0.418
##  5 RetinaNet     Untargeted  obje~   -0.403     0.079     -5.13   0       -0.558
##  6 Faster R-CNN  Vanishing   obje~   -0.618     0.104     -5.96   0       -0.823
##  7 Faster R-CNN  Mislabeling obje~   -0.758     0.131     -5.77   0       -1.02
##  8 Faster R-CNN  Untargeted  obje~   -0.296     0.08      -3.72   0       -0.452
##  9 Cascade R-CNN Vanishing   obje~   -0.779     0.097     -8.00   0       -0.971
## 10 Cascade R-CNN Mislabeling obje~   -0.616     0.11      -5.59   0       -0.833
## 11 Cascade R-CNN Untargeted  obje~   -0.328     0.089     -3.70   0       -0.502
## # i 1 more variable: conf.high <dbl>
```

```
ext_sig(reg_est, "neg", "bbox_dist")
```

```
## ----------bbox_dist----------
## Total 15 predictors:
## 15 (100%) significant;
## 15 (100%) neg
```

```
## # A tibble: 15 x 9
## # Groups:   model_name, loss_target [15]
##    model_name    loss_target term  estimate std.error statistic p.value conf.low
##    <ord>         <ord>       <chr>    <dbl>     <dbl>     <dbl>   <dbl>    <dbl>
##  1 YOLOv3        Vanishing   bbox~    -9.62     0.49      -19.6       0   -10.6
##  2 YOLOv3        Mislabeling bbox~    -7.95     0.43      -18.5       0    -8.80
##  3 YOLOv3        Untargeted  bbox~   -10.3      0.71      -14.5       0   -11.7
##  4 SSD           Vanishing   bbox~   -13.0      0.533     -24.4       0   -14.0
```

```
##  5 SSD           Mislabeling bbox~   -11.7      0.553     -21.2       0   -12.8
##  6 SSD           Untargeted  bbox~   -12.6      0.597     -21.2       0   -13.8
##  7 RetinaNet     Vanishing   bbox~   -28.4      1.62      -17.5       0   -31.6
##  8 RetinaNet     Mislabeling bbox~   -28.6      2.39      -12.0       0   -33.5
##  9 RetinaNet     Untargeted  bbox~   -11.3      0.818     -13.8       0   -12.9
## 10 Faster R-CNN  Vanishing   bbox~   -27.2      1.89      -14.4       0   -31.0
## 11 Faster R-CNN  Mislabeling bbox~   -22.8      2.12      -10.8       0   -27.1
## 12 Faster R-CNN  Untargeted  bbox~   -11.4      0.779     -14.7       0   -13.0
## 13 Cascade R-CNN Vanishing   bbox~   -29.1      1.85      -15.7       0   -32.8
## 14 Cascade R-CNN Mislabeling bbox~   -31.1      2.39      -13.0       0   -36.0
## 15 Cascade R-CNN Untargeted  bbox~   -17.3      1.15      -15.1       0   -19.6
## # i 1 more variable: conf.high <dbl>
```

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

```
## ----------bbox_size_perturb----------
## 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
##    <ord>         <ord>       <chr>    <dbl>     <dbl>     <dbl>   <dbl>    <dbl>
##  1 YOLOv3        Vanishing   bbox~    16.1      0.963     16.8        0    14.3
##  2 YOLOv3        Mislabeling bbox~     8.28     0.521     15.9        0     7.28
##  3 YOLOv3        Untargeted  bbox~     3.02     0.291     10.4        0     2.46
##  4 SSD           Vanishing   bbox~     5.32     0.378     14.1        0     4.59
##  5 SSD           Mislabeling bbox~     6.65     0.403     16.5        0     5.87
##  6 SSD           Untargeted  bbox~     3.26     0.291     11.2        0     2.69
##  7 RetinaNet     Vanishing   bbox~     3.45     0.36       9.59       0     2.76
##  8 RetinaNet     Mislabeling bbox~     2.03     0.412      4.93       0     1.22
##  9 RetinaNet     Untargeted  bbox~     3.66     0.292     12.5        0     3.09
## 10 Faster R-CNN  Vanishing   bbox~     3.37     0.388      8.67       0     2.61
## 11 Faster R-CNN  Mislabeling bbox~     2.00     0.412      4.86       0     1.19
## 12 Faster R-CNN  Untargeted  bbox~     3.75     0.304     12.3        0     3.16
## 13 Cascade R-CNN Vanishing   bbox~     5.75     0.446     12.9        0     4.89
## 14 Cascade R-CNN Mislabeling bbox~     3.18     0.381      8.35       0     2.44
## 15 Cascade R-CNN Untargeted  bbox~     2.75     0.298      9.22       0     2.17
## # i 1 more variable: conf.high <dbl>
```

```
ext_sig(reg_est, "both", "bbox_dist:bbox_size_perturb")
```

```
## ----------bbox_dist:bbox_size_perturb----------
## Total 15 predictors:
## 10 (67%) significant;
## 10 (67%) both
```

```
## # A tibble: 10 x 9
## # Groups:   model_name, loss_target [10]
##    model_name    loss_target term  estimate std.error statistic p.value conf.low
##    <ord>         <ord>       <chr>    <dbl>     <dbl>     <dbl>   <dbl>    <dbl>
##  1 YOLOv3        Vanishing   bbox~   -39.0       5.28     -7.39   0       -49.5
##  2 YOLOv3        Untargeted  bbox~    10.2       2.62      3.90   0         5.10
##  3 SSD           Mislabeling bbox~    -9.85      2.82     -3.50   0       -15.4
##  4 SSD           Untargeted  bbox~     7.14      2.45      2.92   0.004     2.34
```

```
##  5 RetinaNet     Untargeted  bbox~    26.9       2.76      9.75   0        21.6
##  6 Faster R-CNN  Vanishing   bbox~   -19.8       7.38     -2.68   0.007   -34.5
##  7 Faster R-CNN  Untargeted  bbox~    27.4       2.83      9.70   0        22.0
##  8 Cascade R-CNN Vanishing   bbox~   -55.9       8.60     -6.49   0       -73.1
##  9 Cascade R-CNN Mislabeling bbox~   -24.5       9.16     -2.67   0.008   -42.6
## 10 Cascade R-CNN Untargeted  bbox~    22.9       3.29      6.97   0        16.5
## # i 1 more variable: conf.high <dbl>
```

```
dist_lab <- "Perturb-Target Distance"
size_lab <- "Perturb Box Size"

pred_name <- glue("object (versus non-object), with {dist_lab} and {size_lab} as covariates, both relati
main_pt <- "perturbing an object (in the randomized attack) rather than a non-object (in the deliberate

tab_cap <- glue("We combined the data in the randomized and deliberate attack experiments to run a logis

print_statistics(reg_est, tab_cap)
```

Table 2: We combined the data in the randomized and deliberate attack experiments to run a logistic model regressing success against object (versus non-object), with perturb-target distance and perturb box size as covariates, both relative to image width or height. The "object" term codes object as 1 and non-object as 0. Perturbing an object (in the randomized attack) rather than a non-object (in the deliberate attack) significantly decreases success rates for all model and attack combinations, after controlling for perturb sizes and perturb-target distances. Table headers are explained in Appendix **??**.

Group	Regression							
Attack	term	sig	estimate	std.error	statistic	p.value	conf.low	conf.high
YOLOv3								
Vanishing	object	*	-0.537	0.069	-7.786	0.000	-0.673	-0.402
	distance	*	-9.619	0.490	-19.631	0.000	-10.594	-8.673
	size	*	16.138	0.963	16.761	0.000	14.301	18.075
	distance * size	*	-38.994	5.279	-7.387	0.000	-49.534	-28.837
Mislabeling	object	*	-0.622	0.064	-9.731	0.000	-0.747	-0.497
	distance	*	-7.946	0.430	-18.471	0.000	-8.802	-7.116
	size	*	8.275	0.521	15.875	0.000	7.275	9.319
	distance * size		-5.788	3.262	-1.775	0.076	-12.240	0.551
Untargeted	object	*	-0.776	0.077	-10.107	0.000	-0.928	-0.626
	distance	*	-10.294	0.710	-14.502	0.000	-11.713	-8.930
	size	*	3.025	0.291	10.388	0.000	2.457	3.599
	distance * size	*	10.204	2.615	3.902	0.000	5.096	15.352
SSD								
Vanishing	object	*	0.325	0.064	5.072	0.000	0.200	0.451
	distance	*	-12.970	0.533	-24.350	0.000	-14.031	-11.943
	size	*	5.319	0.378	14.081	0.000	4.590	6.071
	distance * size		1.653	2.648	0.624	0.533	-3.560	6.824
Mislabeling	object		-0.101	0.064	-1.585	0.113	-0.226	0.024

	distance	*	-11.732	0.553	-21.216	0.000	-12.834	-10.666
	size	*	6.651	0.403	16.492	0.000	5.873	7.454
	distance * size	*	-9.854	2.818	-3.497	0.000	-15.407	-4.359
Untargeted	object		0.027	0.064	0.424	0.672	-0.098	0.152
	distance	*	-12.646	0.597	-21.177	0.000	-13.838	-11.497
	size	*	3.258	0.291	11.201	0.000	2.693	3.834
	distance * size	*	7.145	2.448	2.919	0.004	2.344	11.942
RetinaNet								
Vanishing	object	*	-0.251	0.085	-2.953	0.003	-0.418	-0.085
	distance	*	-28.371	1.624	-17.466	0.000	-31.631	-25.264
	size	*	3.453	0.360	9.591	0.000	2.755	4.167
	distance * size		-5.791	5.990	-0.967	0.334	-17.676	5.813
Mislabeling	object		-0.164	0.113	-1.447	0.148	-0.388	0.057
	distance	*	-28.622	2.391	-11.973	0.000	-33.480	-24.110
	size	*	2.030	0.412	4.926	0.000	1.224	2.840
	distance * size		-6.022	8.891	-0.677	0.498	-23.711	11.158
Untargeted	object	*	-0.403	0.079	-5.130	0.000	-0.558	-0.250
	distance	*	-11.268	0.818	-13.768	0.000	-12.910	-9.702
	size	*	3.662	0.292	12.542	0.000	3.092	4.237
	distance * size	*	26.886	2.757	9.753	0.000	21.555	32.364
Faster R-CNN								
Vanishing	object	*	-0.618	0.104	-5.964	0.000	-0.823	-0.416
	distance	*	-27.236	1.889	-14.422	0.000	-31.047	-23.643
	size	*	3.369	0.388	8.671	0.000	2.614	4.137
	distance * size	*	-19.812	7.379	-2.685	0.007	-34.469	-5.530
Mislabeling	object	*	-0.758	0.131	-5.767	0.000	-1.019	-0.504
	distance	*	-22.755	2.115	-10.757	0.000	-27.063	-18.771
	size	*	2.001	0.412	4.857	0.000	1.194	2.810
	distance * size		-14.270	8.311	-1.717	0.086	-30.831	1.768
Untargeted	object	*	-0.296	0.080	-3.719	0.000	-0.452	-0.140
	distance	*	-11.447	0.779	-14.701	0.000	-13.004	-9.953
	size	*	3.748	0.304	12.322	0.000	3.155	4.347
	distance * size	*	27.445	2.829	9.703	0.000	21.965	33.056
Cascade R-CNN								
Vanishing	object	*	-0.779	0.097	-7.999	0.000	-0.971	-0.589
	distance	*	-29.119	1.854	-15.710	0.000	-32.850	-25.584
	size	*	5.752	0.446	12.907	0.000	4.894	6.642
	distance * size	*	-55.876	8.604	-6.494	0.000	-73.094	-39.336
Mislabeling	object	*	-0.616	0.110	-5.592	0.000	-0.833	-0.401
	distance	*	-31.146	2.387	-13.046	0.000	-35.990	-26.630
	size	*	3.180	0.381	8.347	0.000	2.438	3.933

	distance * size	*	-24.457	9.159	-2.670	0.008	-42.647	-6.724
Untargeted	object	*	-0.328	0.089	-3.701	0.000	-0.502	-0.155
	distance	*	-17.329	1.148	-15.089	0.000	-19.637	-15.134
	size	*	2.749	0.298	9.221	0.000	2.166	3.335
	distance * size	*	22.929	3.289	6.972	0.000	16.523	29.419