Arbitrary Experiment

```r
library(conflicted)

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

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

```r
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
```

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

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

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

```r
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 ve
##
## 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

```r
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               <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> 52, 52, 52, 52, 53, 52, 52, 52, 52, 53, 52, 52~
## $ attack_count           <dbl> 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50~
## $ success_count          <dbl> 10, 10, 5, 2, 21, 13, 13, 12, 7, 13, 15, 16, 1~
## $ vanish_count           <dbl> 3, 2, 1, 1, 10, 10, 13, 8, 5, 11, 14, 14, 18, ~
## $ mislabel_count         <dbl> 7, 8, 4, 1, 11, 3, 0, 4, 2, 2, 1, 2, 1, 1, 0, ~
## $ mislabel_intended_count <dbl> 7, 8, 4, 1, 11, 0, 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~
```

```r
# 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                <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> 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52~
## $ attack_count            <dbl> 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50~
## $ success_count           <dbl> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10~
## $ vanish_count            <dbl> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3~
## $ mislabel_count          <dbl> 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7~
## $ mislabel_intended_count <dbl> 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7~
## $ 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~
```

```r
# 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~    -6.05      1.23     -4.93       0    -8.47
##  2 YOLOv3        Mislabeling bbox~    -7.15      1.23     -5.81       0    -9.59
##  3 YOLOv3        Untargeted  bbox~    -9.32      1.52     -6.15       0   -12.3
##  4 SSD           Vanishing   bbox~   -10.4       1.55     -6.71       0   -13.5
##  5 SSD           Mislabeling bbox~    -8.00      1.70     -4.71       0   -11.4
##  6 SSD           Untargeted  bbox~    -9.78      1.87     -5.23       0   -13.5
##  7 RetinaNet     Vanishing   bbox~   -23.0       3.08     -7.48       0   -29.3
##  8 RetinaNet     Mislabeling bbox~   -22.5       4.24     -5.32       0   -31.3
##  9 RetinaNet     Untargeted  bbox~   -23.5       2.44     -9.64       0   -28.4
## 10 Faster R-CNN  Vanishing   bbox~   -31.8       4.00     -7.95       0   -39.9
## 11 Faster R-CNN  Mislabeling bbox~   -28.0       4.33     -6.47       0   -36.9
## 12 Faster R-CNN  Untargeted  bbox~   -29.7       2.76    -10.8        0   -35.3
## 13 Cascade R-CNN Vanishing   bbox~   -33.2       4.28     -7.76       0   -41.9
## 14 Cascade R-CNN Mislabeling bbox~   -34.8       5.23     -6.65       0   -45.6
## 15 Cascade R-CNN Untargeted  bbox~   -45.7       4.12    -11.1        0   -54.0
## # 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~    8.24      0.53      15.6     0        7.23
##  2 YOLOv3        Mislabeling bbox~    5.89      0.395     14.9     0        5.13
##  3 YOLOv3        Untargeted  bbox~    2.06      0.285      7.24    0        1.51
##  4 SSD           Vanishing   bbox~    4.74      0.318     14.9     0        4.12
##  5 SSD           Mislabeling bbox~    6.06      0.345     17.6     0        5.40
##  6 SSD           Untargeted  bbox~    3.80      0.314     12.1     0        3.19
##  7 RetinaNet     Vanishing   bbox~    2.58      0.345      7.49    0        1.91
##  8 RetinaNet     Mislabeling bbox~    1.26      0.419      3.01    0.003    0.443
##  9 RetinaNet     Untargeted  bbox~    2.53      0.334      7.57    0        1.88
## 10 Faster R-CNN  Vanishing   bbox~    2.08      0.36       5.77    0        1.38
## 11 Faster R-CNN  Mislabeling bbox~    0.955     0.395      2.42    0.016    0.185
## 12 Faster R-CNN  Untargeted  bbox~    1.49      0.312      4.78    0        0.886
## 13 Cascade R-CNN Vanishing   bbox~    3.93      0.405      9.71    0        3.14
## 14 Cascade R-CNN Mislabeling bbox~    1.85      0.395      4.70    0        1.08
## 15 Cascade R-CNN Untargeted  bbox~    0.675     0.327      2.06   0.039    0.036
## # i 1 more variable: conf.high <dbl>
```

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

```
## ----------bbox_dist:bbox_len----------
## Total 15 predictors:
## 7 (47%) significant;
## 7 (47%) both
```

```
## # A tibble: 7 x 9
## # Groups:   model_name, loss_target [7]
##   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_~   -18.2       3.54     -5.14   0        -25.2
## 2 SSD           Mislabeling bbox_~   -12.7       3.35     -3.77   0        -19.2
## 3 SSD           Untargeted  bbox_~    -7.44      3.64     -2.05   0.041    -14.5
## 4 RetinaNet     Untargeted  bbox_~    37.7       4.19      8.99   0         29.6
## 5 Faster R-CNN  Untargeted  bbox_~    36.7       4.55      8.07   0         27.9
## 6 Cascade R-CNN Vanishing   bbox_~   -22.5       8.93     -2.52   0.012    -40.0
## 7 Cascade R-CNN Untargeted  bbox_~    47.7       6.64      7.19   0         35.0
## # 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	*	-6.047	1.227	-4.928	0.000	-8.472	-3.660
	length	*	8.243	0.530	15.558	0.000	7.227	9.305
	distance * length	*	-18.211	3.543	-5.140	0.000	-25.189	-11.292
Mislabeling	distance	*	-7.151	1.231	-5.810	0.000	-9.588	-4.761
	length	*	5.888	0.395	14.922	0.000	5.126	6.674
	distance * length		-3.239	3.100	-1.045	0.296	-9.296	2.862
Untargeted	distance	*	-9.320	1.515	-6.153	0.000	-12.343	-6.401
	length	*	2.063	0.285	7.245	0.000	1.508	2.624
	distance * length		4.340	2.943	1.475	0.140	-1.392	10.150
SSD								

	Vanishing	distance	*	-10.417	1.552	-6.711	0.000	-13.513	-7.424
		length	*	4.737	0.318	14.882	0.000	4.120	5.368
		distance * length		-3.353	3.072	-1.091	0.275	-9.345	2.705
	Mislabeling	distance	*	-7.996	1.697	-4.712	0.000	-11.385	-4.729
		length	*	6.065	0.345	17.570	0.000	5.397	6.750
		distance * length	*	-12.651	3.354	-3.772	0.000	-19.201	-6.047
	Untargeted	distance	*	-9.777	1.868	-5.233	0.000	-13.530	-6.201
		length	*	3.798	0.314	12.094	0.000	3.188	4.419
		distance * length	*	-7.443	3.635	-2.048	0.041	-14.527	-0.268
RetinaNet									
	Vanishing	distance	*	-23.008	3.077	-7.477	0.000	-29.253	-17.194
		length	*	2.583	0.345	7.491	0.000	1.912	3.264
		distance * length		-10.769	6.353	-1.695	0.090	-23.153	1.757
	Mislabeling	distance	*	-22.522	4.237	-5.316	0.000	-31.273	-14.667
		length	*	1.261	0.419	3.007	0.003	0.443	2.087
		distance * length		1.459	8.334	0.175	0.861	-14.680	18.011
	Untargeted	distance	*	-23.500	2.437	-9.643	0.000	-28.382	-18.828
		length	*	2.528	0.334	7.571	0.000	1.880	3.189
		distance * length	*	37.697	4.191	8.994	0.000	29.615	46.048
Faster R-CNN									
	Vanishing	distance	*	-31.756	3.996	-7.947	0.000	-39.875	-24.217
		length	*	2.075	0.360	5.770	0.000	1.375	2.785
		distance * length		-0.099	7.820	-0.013	0.990	-15.305	15.352
	Mislabeling	distance	*	-28.038	4.331	-6.474	0.000	-36.927	-19.955
		length	*	0.955	0.395	2.419	0.016	0.185	1.734
		distance * length		10.044	8.211	1.223	0.221	-5.864	26.342
	Untargeted	distance	*	-29.741	2.761	-10.770	0.000	-35.304	-24.477
		length	*	1.494	0.312	4.783	0.000	0.886	2.111
		distance * length	*	36.707	4.548	8.071	0.000	27.946	45.780
Cascade R-CNN									
	Vanishing	distance	*	-33.193	4.280	-7.755	0.000	-41.863	-25.092
		length	*	3.929	0.405	9.706	0.000	3.145	4.732
		distance * length	*	-22.519	8.925	-2.523	0.012	-39.964	-4.967
	Mislabeling	distance	*	-34.815	5.234	-6.652	0.000	-45.560	-25.047
		length	*	1.853	0.395	4.698	0.000	1.085	2.632
		distance * length		-2.173	10.288	-0.211	0.833	-22.101	18.246
	Untargeted	distance	*	-45.652	4.120	-11.080	0.000	-53.998	-37.841
		length	*	0.675	0.327	2.061	0.039	0.036	1.320
		distance * length	*	47.723	6.636	7.191	0.000	34.958	60.993

```
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> 0.2227890, 1.8349264, 3.4470637, 5.0592010, 0.2076889, 1.8~
## $ .se.fit     <dbl> 0.10111954, 0.08791323, 0.15870232, 0.25019639, 0.09975179~
## $ success     <dbl> 0.5554680, 0.8623476, 0.9691435, 0.9936894, 0.5517364, 0.8~
## $ ul          <dbl> 0.6037185, 0.8815548, 0.9772042, 0.9961260, 0.5994568, 0.8~
## $ ll          <dbl> 0.5061484, 0.8405889, 0.9583538, 0.9897362, 0.5030438, 0.8~
```

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

arb_cap
```

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

```
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(
```

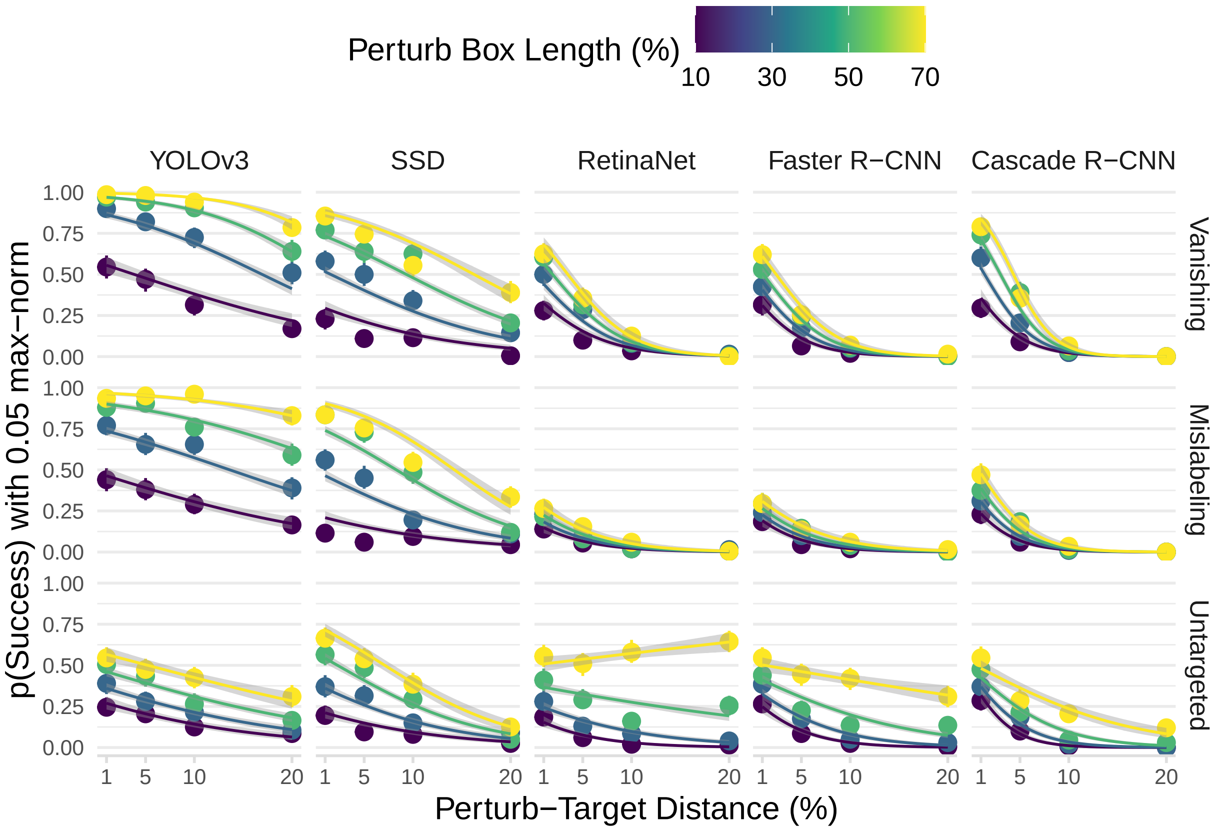

Figure 1: Perturbing an arbitrary region obfuscates intent with increased success for all models and attacks even with 0.05 max-norm: We implement intent obfuscating attack by perturbing an arbitrary non-overlapping square region to disrupt a randomly selected target object at various lengths and distances. 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 ??.

```
  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(s
  scale_color_viridis_c(name = glue("{len_lab} (%)"), breaks = unique(success_expanded_data$bbox_len), 
```

```
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~
```

```
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~
```

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

```
# 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 4 warnings in `mutate()`.
## The first warning was:
## i In argument: `mod = list(model(data))`.
## i In row 8.
## Caused by warning:
## ! glm.fit: fitted probabilities numerically 0 or 1 occurred
## i Run `dplyr::last_dplyr_warnings()` to see the 3 remaining warnings.
```

```
## Warning: There were 234 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 233 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:
## 10 (67%) significant;
## 6 (40%) neg

## # A tibble: 6 x 9
## # Groups:   model_name, loss_target [6]
##   model_name    loss_target term   estimate std.error statistic p.value conf.low
##   <ord>         <ord>       <chr>     <dbl>     <dbl>     <dbl>   <dbl>    <dbl>
## 1 YOLOv3        Mislabeling object   -0.254     0.064     -3.98   0        -0.379
## 2 YOLOv3        Untargeted  object   -0.533     0.078     -6.81   0        -0.687
## 3 Faster R-CNN  Vanishing   object   -0.478     0.105     -4.53   0        -0.686
## 4 Faster R-CNN  Mislabeling object   -0.636     0.133     -4.78   0        -0.9
## 5 Cascade R-CNN Vanishing   object   -0.437     0.099     -4.41   0        -0.632
## 6 Cascade R-CNN Mislabeling object   -0.314     0.112     -2.80   0.005    -0.535
## # 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~    -9.45     0.482     -19.6       0    -10.4
##  2 YOLOv3        Mislabeling bbox~    -8.05     0.428     -18.8       0     -8.90
##  3 YOLOv3        Untargeted  bbox~   -10.8      0.73      -14.8       0    -12.2
##  4 SSD           Vanishing   bbox~   -13.7      0.574     -23.9       0    -14.9
##  5 SSD           Mislabeling bbox~   -11.8      0.585     -20.1       0    -13.0
##  6 SSD           Untargeted  bbox~   -12.9      0.654     -19.7       0    -14.2
##  7 RetinaNet     Vanishing   bbox~   -30.8      1.73      -17.8       0    -34.2
##  8 RetinaNet     Mislabeling bbox~   -33.5      2.68      -12.5       0    -38.9
##  9 RetinaNet     Untargeted  bbox~   -13.2      0.915     -14.4       0    -15.1
## 10 Faster R-CNN  Vanishing   bbox~   -31.8      2.12      -15.0       0    -36.1
## 11 Faster R-CNN  Mislabeling bbox~   -26.1      2.30      -11.3       0    -30.8
## 12 Faster R-CNN  Untargeted  bbox~   -13.1      0.864     -15.1       0    -14.8
## 13 Cascade R-CNN Vanishing   bbox~   -32.3      2.08      -15.5       0    -36.5
## 14 Cascade R-CNN Mislabeling bbox~   -32.4      2.53      -12.8       0    -37.6
## 15 Cascade R-CNN Untargeted  bbox~   -19.0      1.27      -15.0       0    -21.6
## # i 1 more variable: conf.high <dbl>
```

```r
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~   16.4       0.9        18.2    0        14.6
##  2 YOLOv3        Mislabeling bbox~    7.94      0.471      16.8    0         7.03
##  3 YOLOv3        Untargeted  bbox~    2.09      0.286       7.32   0         1.53
##  4 SSD           Vanishing   bbox~    5.48      0.339      16.2    0         4.82
##  5 SSD           Mislabeling bbox~    6.62      0.346      19.1    0         5.95
##  6 SSD           Untargeted  bbox~    3.60      0.288      12.5    0         3.04
##  7 RetinaNet     Vanishing   bbox~    2.62      0.343       7.63   0         1.95
##  8 RetinaNet     Mislabeling bbox~    0.871     0.419       2.08   0.038     0.051
##  9 RetinaNet     Untargeted  bbox~    3.03      0.295      10.3    0         2.46
## 10 Faster R-CNN  Vanishing   bbox~    2.43      0.382       6.37   0         1.69
## 11 Faster R-CNN  Mislabeling bbox~    0.908     0.419       2.17   0.03      0.086
## 12 Faster R-CNN  Untargeted  bbox~    2.91      0.302       9.64   0         2.32
## 13 Cascade R-CNN Vanishing   bbox~    5.21      0.426      12.2    0         4.39
## 14 Cascade R-CNN Mislabeling bbox~    2.19      0.381       5.74   0         1.44
## 15 Cascade R-CNN Untargeted  bbox~    2.10      0.308       6.84   0         1.50
## # i 1 more variable: conf.high <dbl>
```

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

```
## ----------bbox_dist:bbox_size_perturb----------
## Total 15 predictors:
## 9 (60%) significant;
## 9 (60%) both

## # A tibble: 9 x 9
## # Groups:   model_name, loss_target [9]
##   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_~    -43.8      4.94     -8.87   0        -53.6
## 2 YOLOv3        Mislabeling bbox_~     -7.1      3.00     -2.36   0.018   -13.0
## 3 YOLOv3        Untargeted  bbox_~     12.5      2.65      4.73   0          7.34
## 4 SSD           Mislabeling bbox_~    -10.2      2.67     -3.83   0        -15.5
## 5 RetinaNet     Mislabeling bbox_~     23.2      8.51      2.73   0.006      6.31
## 6 RetinaNet     Untargeted  bbox_~     33.6      2.90     11.6    0         28.0
## 7 Faster R-CNN  Untargeted  bbox_~     25.7      2.8       9.16   0         20.2
## 8 Cascade R-CNN Vanishing   bbox_~    -42.5      8.79     -4.84   0        -60.1
## 9 Cascade R-CNN Untargeted  bbox_~     19.6      3.77      5.19   0         12.2
## # i 1 more variable: conf.high <dbl>
```

```r
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 relat
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 logi

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		-0.126	0.069	-1.829	0.067	-0.260	0.009
	distance	*	-9.446	0.482	-19.592	0.000	-10.405	-8.515
	size	*	16.353	0.900	18.179	0.000	14.634	18.161
	distance * size	*	-43.789	4.938	-8.867	0.000	-53.633	-34.267
Mislabeling	object	*	-0.254	0.064	-3.985	0.000	-0.379	-0.129
	distance	*	-8.051	0.428	-18.833	0.000	-8.902	-7.226
	size	*	7.939	0.471	16.845	0.000	7.034	8.882
	distance * size	*	-7.100	3.004	-2.364	0.018	-13.029	-1.249
Untargeted	object	*	-0.533	0.078	-6.807	0.000	-0.687	-0.380
	distance	*	-10.771	0.730	-14.752	0.000	-12.232	-9.370
	size	*	2.091	0.286	7.317	0.000	1.532	2.653
	distance * size	*	12.506	2.646	4.726	0.000	7.335	17.711
SSD								
Vanishing	object	*	1.143	0.069	16.606	0.000	1.009	1.278
	distance	*	-13.732	0.574	-23.911	0.000	-14.878	-12.627
	size	*	5.475	0.339	16.160	0.000	4.819	6.147
	distance * size		1.736	2.532	0.686	0.493	-3.244	6.686
Mislabeling	object	*	0.887	0.069	12.813	0.000	0.752	1.023
	distance	*	-11.787	0.585	-20.137	0.000	-12.957	-10.663
	size	*	6.622	0.346	19.126	0.000	5.952	7.309
	distance * size	*	-10.236	2.674	-3.829	0.000	-15.506	-5.022
Untargeted	object	*	0.914	0.070	12.977	0.000	0.777	1.053
	distance	*	-12.866	0.654	-19.665	0.000	-14.176	-11.611
	size	*	3.596	0.288	12.503	0.000	3.036	4.164
	distance * size		-0.763	2.732	-0.279	0.780	-6.149	4.566
RetinaNet								
Vanishing	object		-0.034	0.086	-0.395	0.693	-0.203	0.135
	distance	*	-30.751	1.730	-17.775	0.000	-34.225	-27.443

	size	*	2.620	0.343	7.634	0.000	1.953	3.299
	distance * size		9.753	5.793	1.684	0.092	-1.711	21.010
Mislabeling	object		0.037	0.116	0.322	0.747	-0.191	0.264
	distance	*	-33.503	2.676	-12.518	0.000	-38.938	-28.446
	size	*	0.871	0.419	2.080	0.038	0.051	1.693
	distance * size	*	23.197	8.508	2.726	0.006	6.307	39.700
Untargeted	object		-0.043	0.081	-0.522	0.601	-0.202	0.117
	distance	*	-13.217	0.915	-14.441	0.000	-15.053	-11.466
	size	*	3.032	0.295	10.289	0.000	2.456	3.611
	distance * size	*	33.609	2.898	11.599	0.000	28.011	39.372
Faster R-CNN								
Vanishing	object	*	-0.478	0.105	-4.529	0.000	-0.686	-0.272
	distance	*	-31.827	2.118	-15.029	0.000	-36.100	-27.798
	size	*	2.432	0.382	6.368	0.000	1.689	3.186
	distance * size		-3.404	7.606	-0.448	0.654	-18.494	11.337
Mislabeling	object	*	-0.636	0.133	-4.778	0.000	-0.900	-0.378
	distance	*	-26.142	2.304	-11.348	0.000	-30.831	-21.799
	size	*	0.908	0.419	2.168	0.030	0.086	1.730
	distance * size		8.990	7.894	1.139	0.255	-6.721	24.259
Untargeted	object	*	0.272	0.084	3.241	0.001	0.108	0.437
	distance	*	-13.071	0.864	-15.131	0.000	-14.804	-11.418
	size	*	2.907	0.302	9.640	0.000	2.318	3.500
	distance * size	*	25.656	2.800	9.164	0.000	20.216	31.193
Cascade R-CNN								
Vanishing	object	*	-0.437	0.099	-4.409	0.000	-0.632	-0.243
	distance	*	-32.264	2.078	-15.526	0.000	-36.452	-28.306
	size	*	5.213	0.426	12.239	0.000	4.392	6.063
	distance * size	*	-42.522	8.789	-4.838	0.000	-60.059	-25.581
Mislabeling	object	*	-0.314	0.112	-2.803	0.005	-0.535	-0.096
	distance	*	-32.423	2.526	-12.835	0.000	-37.559	-27.654
	size	*	2.189	0.381	5.738	0.000	1.443	2.939
	distance * size		-12.586	9.615	-1.309	0.191	-31.740	5.972
Untargeted	object		-0.075	0.091	-0.825	0.409	-0.255	0.103
	distance	*	-19.039	1.269	-15.008	0.000	-21.594	-16.620
	size	*	2.105	0.308	6.837	0.000	1.503	2.711
	distance * size	*	19.565	3.768	5.192	0.000	12.194	26.975