Validation set vs training set training - Approach two

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.5

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```r
library(ggpubr)
```

```
## Warning: package 'ggpubr' was built under R version 4.0.5
```

This experiment focuses on the question, whether training linear combining method on the same set of data as the neural networks were trained on has any adverse effects to the performance of the ensemble as opposed to training on a different set, not presented to the networks during the training. Experiment was performed with two slightly different approaches.

Approach two Experiment code is in the file half_train_ensembling_experiment.py. Experiment on both CIFAR10 and CIFAR100 datasets. This experiment differs from the previous one in the neural networks training part. In this case, the networks were trained on half of the original training set. The remainder of the training set was extracted as a validation set. This enabled us to train several ensembles on both the training set and the validation set in each replication. Experiment on CIFAR10 was performed in 1 replication and experiment on CIFAR100 in 10 replications. This is due to 10 times more classes in CIFAR100 and thus a need for 10 times larger combiner training set in order to maintain constant 50 samples for class in combiner models training. We found that we are getting statistically significant, but opposing results for different sets of combined networks, therefore we performed the test on all 11 possible subsets (of size at least 2) of the four trained networks.

Approach one is described in the file visualization_base_experiment.

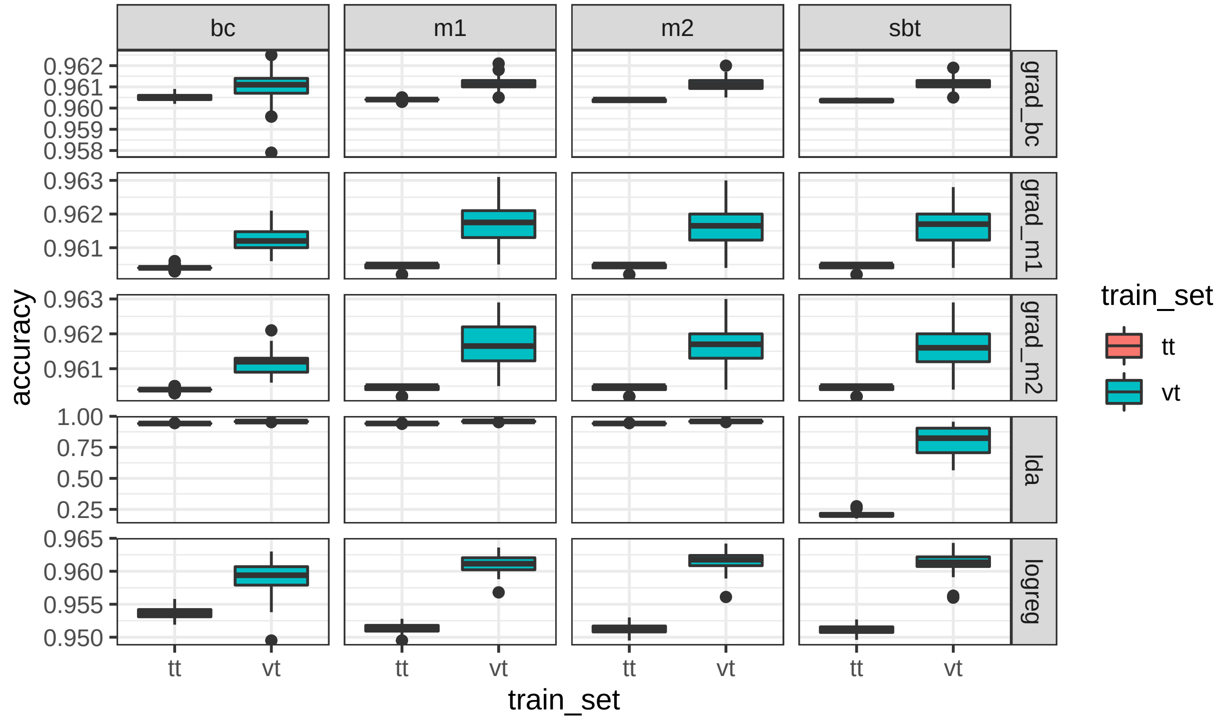
```
metrics <- c("accuracy", "nll", "ece")
metric_names <- c("accuracy", "NLL", "ECE")
metrics_opt <- c("max", "min", "min")
```

CIFAR10

```
net_results_c10 <- read.csv("../data/data_train_val_half_c10/net_metrics.csv")
ens_results_c10 <- read.csv("../data/data_train_val_half_c10/ensemble_metrics.csv")
net_cols <- gsub("-", ".", unique(net_results_c10$network))
```

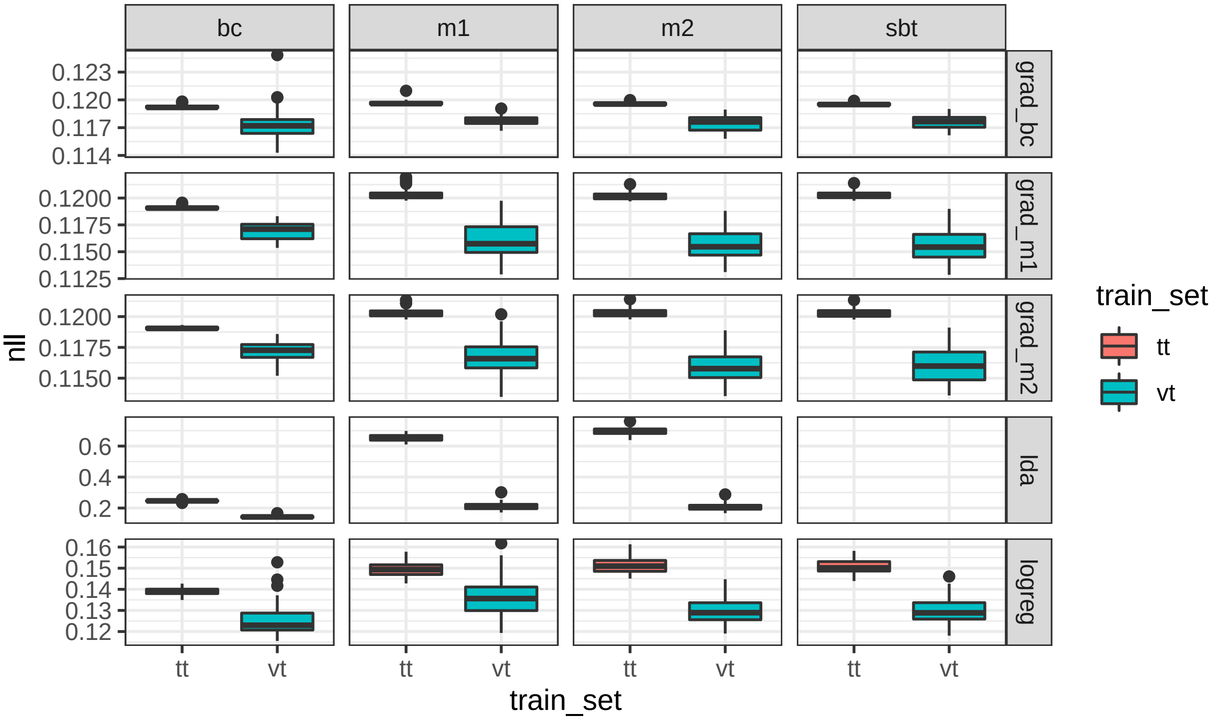
```
for (comb_id in unique(ens_results_c10$combination_id))
{
    cur_comb_ens_results <- ens_results_c10 %>% filter(combination_id == comb_id)
    comb_nets <- gsub("\\.", "_", net_cols[as.logical(cur_comb_ens_results[1, net_cols])])
    for (met_i in seq_along(metrics))
    {
        box_plot <- cur_comb_ens_results  %>% ggplot() +
            geom_boxplot(mapping=aes_string(x = "train_set", y = metrics[met_i], fill = "train_set")) +
            facet_grid(rows=vars(combining_method), cols=vars(coupling_method), scales="free") +
            ggtitle(paste0(
                "CIFAR-10. Metric ", metric_names[met_i],
                " of ensembles with combining method\ntrained on different train sets\nnetworks ",
                paste(comb_nets, collapse = " "))) +
            theme_bw()
        print(box_plot)
    }
}
```

CIFAR−10. Metric accuracy of ensembles with combining method trained on different train sets
networks clip_ViT_B_32_LP densenet121

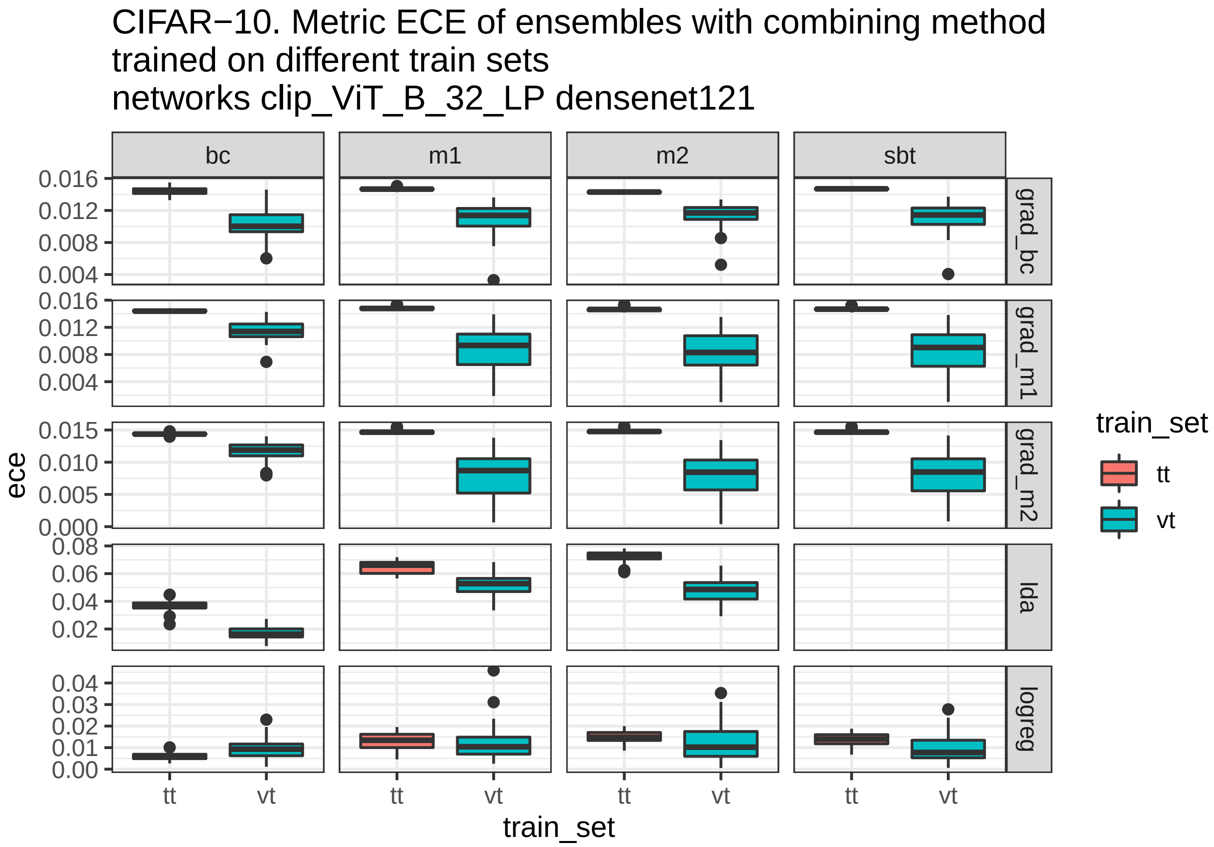


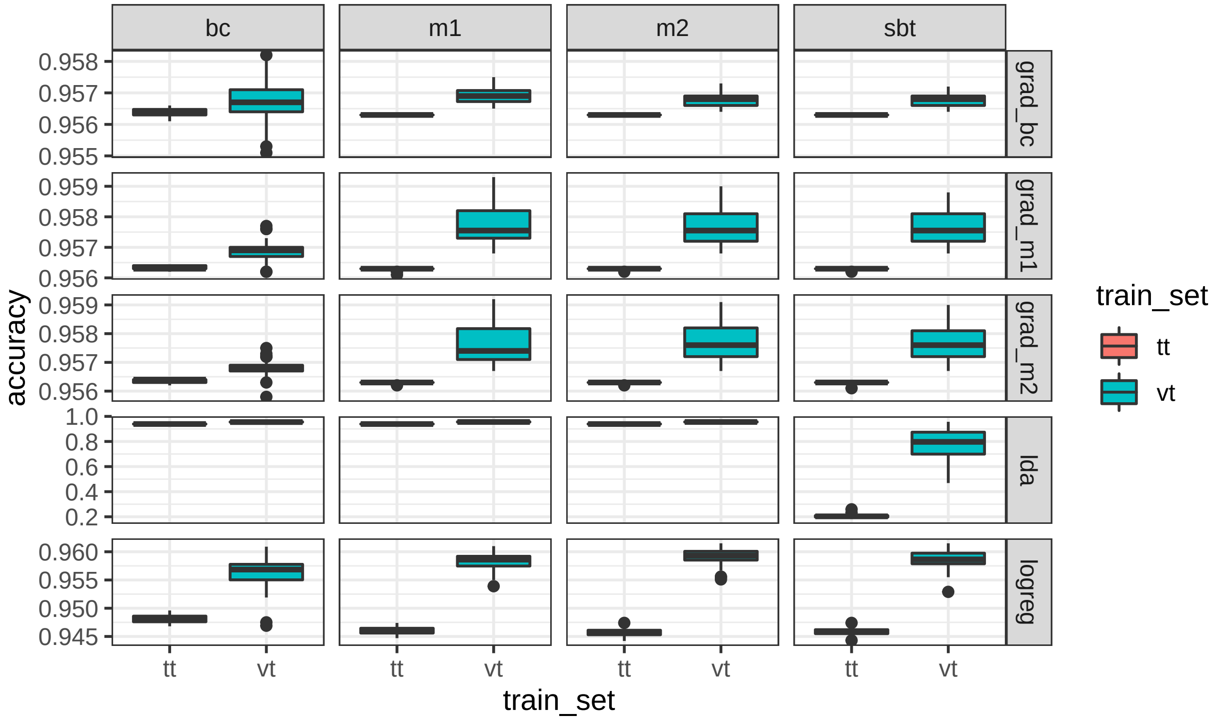
```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```

CIFAR−10. Metric NLL of ensembles with combining method trained on different train sets
networks clip_ViT_B_32_LP densenet121



```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```

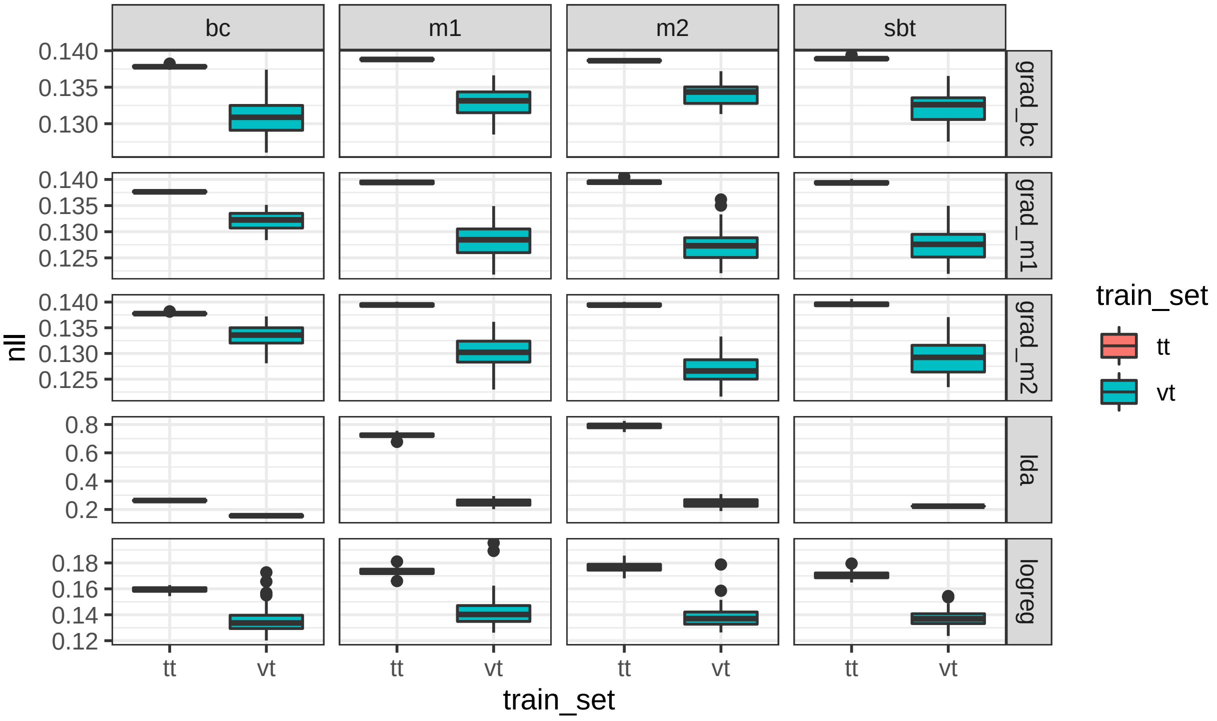

CIFAR−10. Metric ECE of ensembles with combining method
trained on different train sets
networks clip_ViT_B_32_LP densenet121
bc
m1
m2
sbt
grad_bc
grad_m1
grad_m2
lda
logreg
0.016
0.012
0.008
0.004
0.016
0.012
0.008
0.004
0.015
0.010
0.005
0.000
0.08
0.06
0.04
0.02
0.04
0.03
0.02
0.01
0.00
ece
tt
vt
tt
vt
tt
vt
tt
vt
train_set
train_set
tt
vt



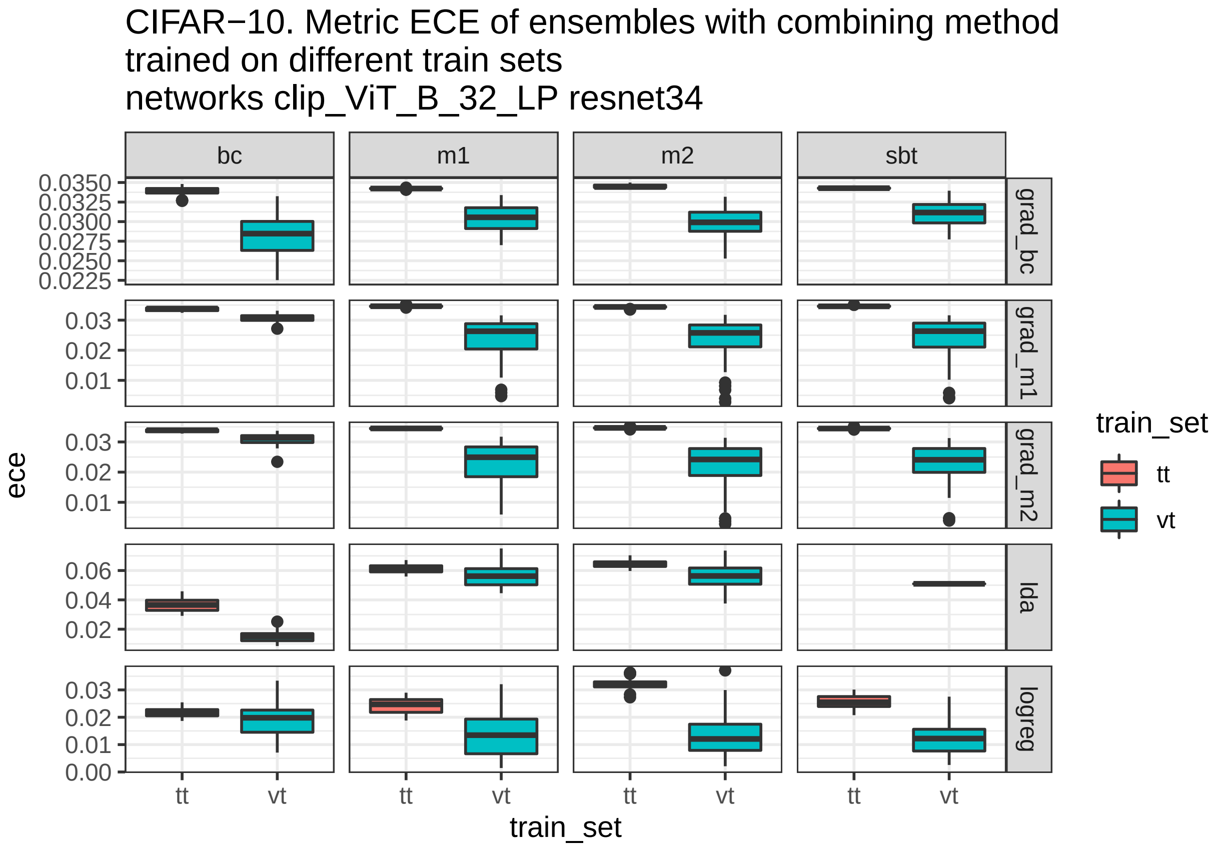
CIFAR−10. Metric accuracy of ensembles with combining method
trained on different train sets
networks clip_ViT_B_32_LP resnet34

```
## Warning: Removed 99 rows containing non-finite values (stat_boxplot).
```

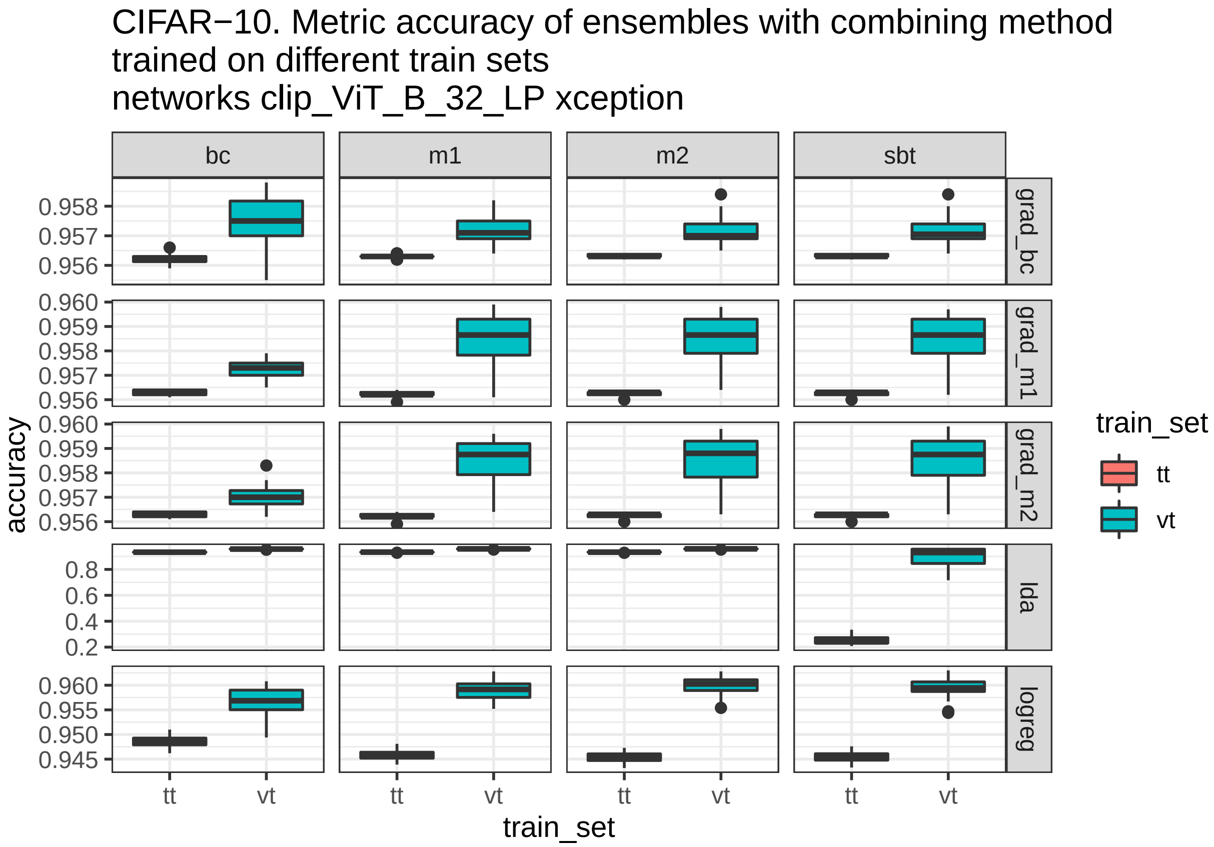
CIFAR−10. Metric NLL of ensembles with combining method
trained on different train sets
networks clip_ViT_B_32_LP resnet34



```
## Warning: Removed 99 rows containing non-finite values (stat_boxplot).
```

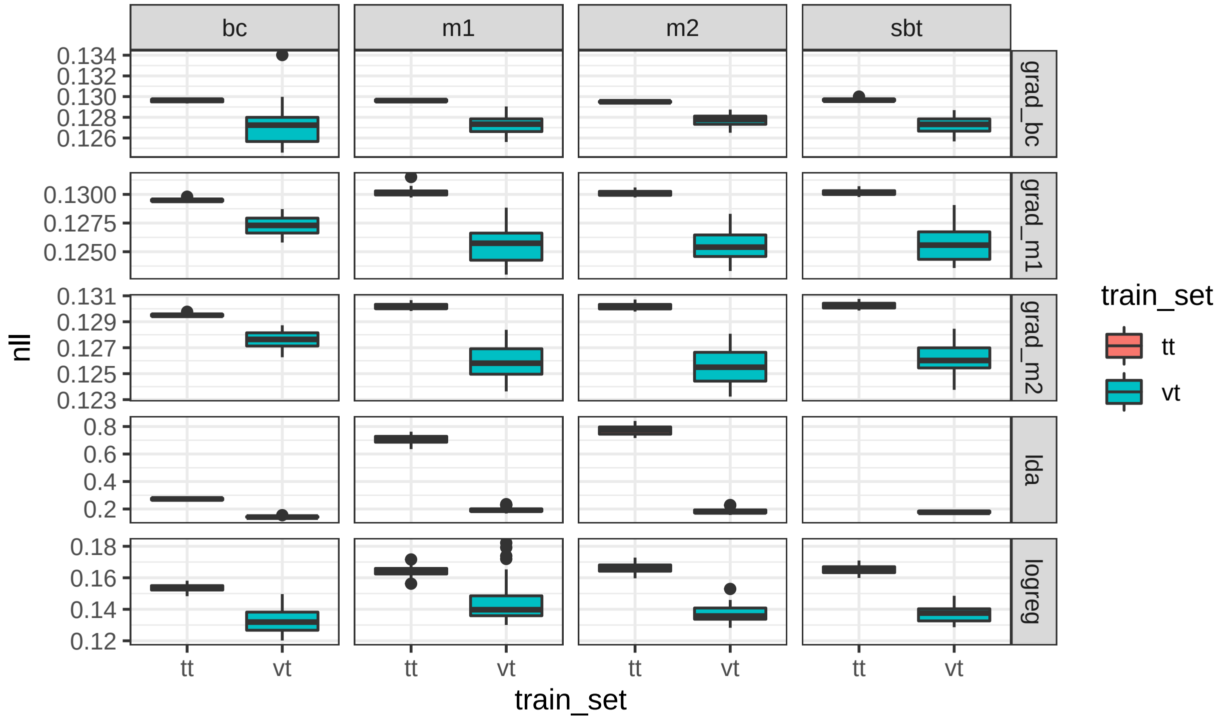


CIFAR−10. Metric ECE of ensembles with combining method trained on different train sets
networks clip_ViT_B_32_LP resnet34

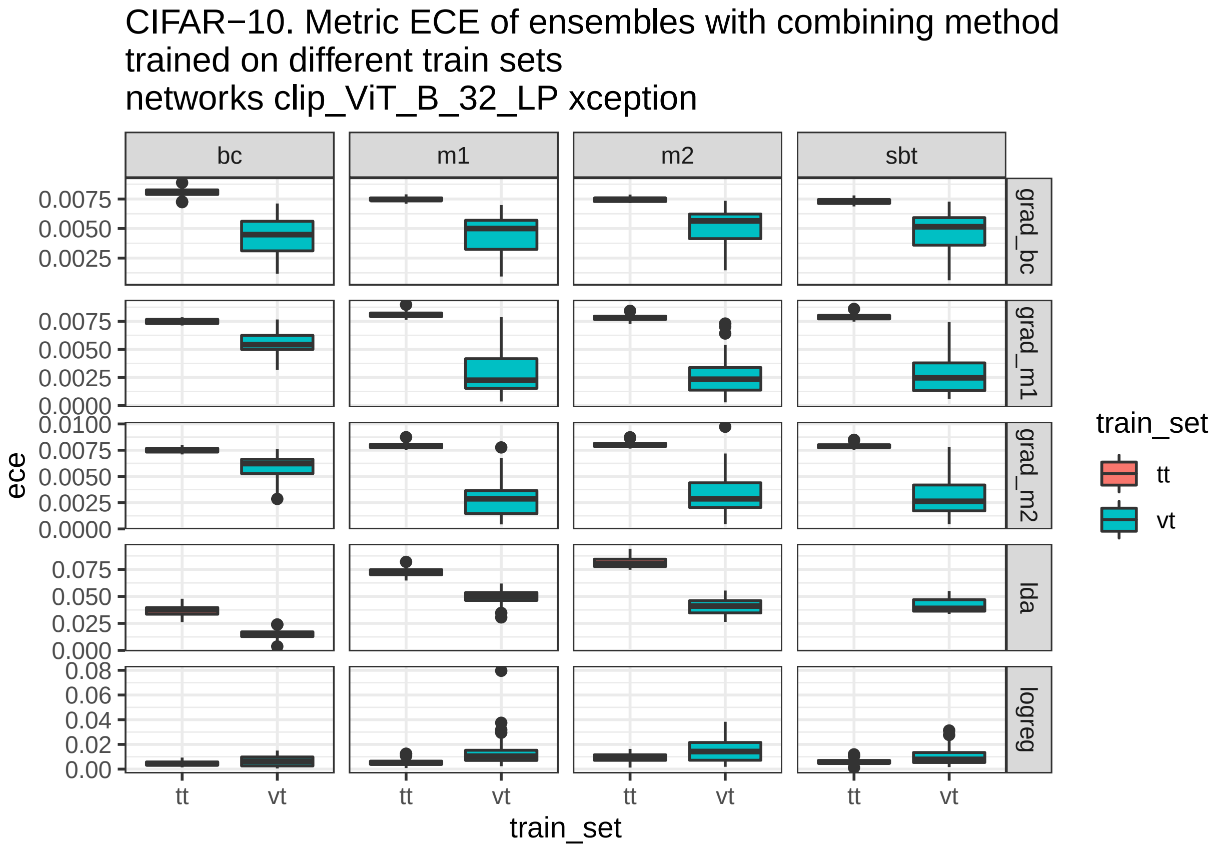


```
## Warning: Removed 97 rows containing non-finite values (stat_boxplot).
```

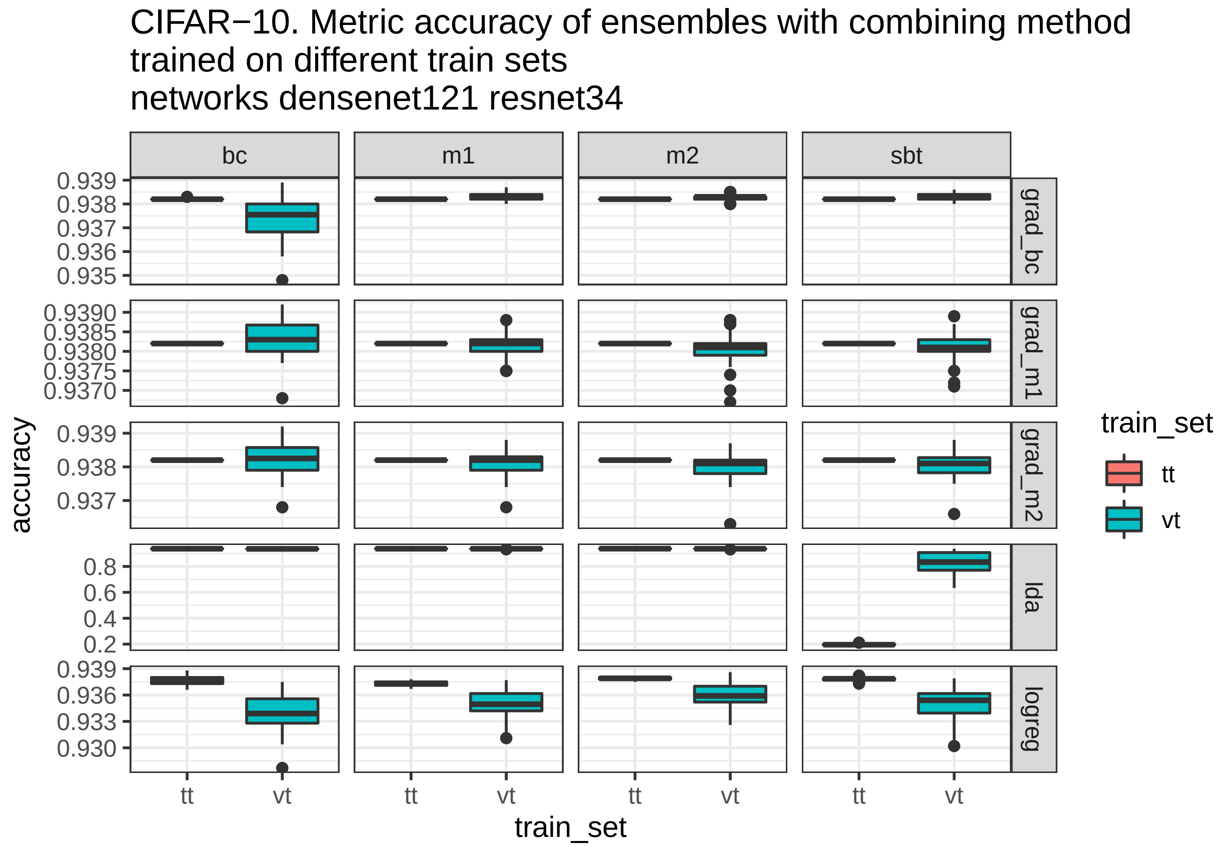
CIFAR−10. Metric NLL of ensembles with combining method
trained on different train sets
networks clip_ViT_B_32_LP xception



```
## Warning: Removed 97 rows containing non-finite values (stat_boxplot).
```

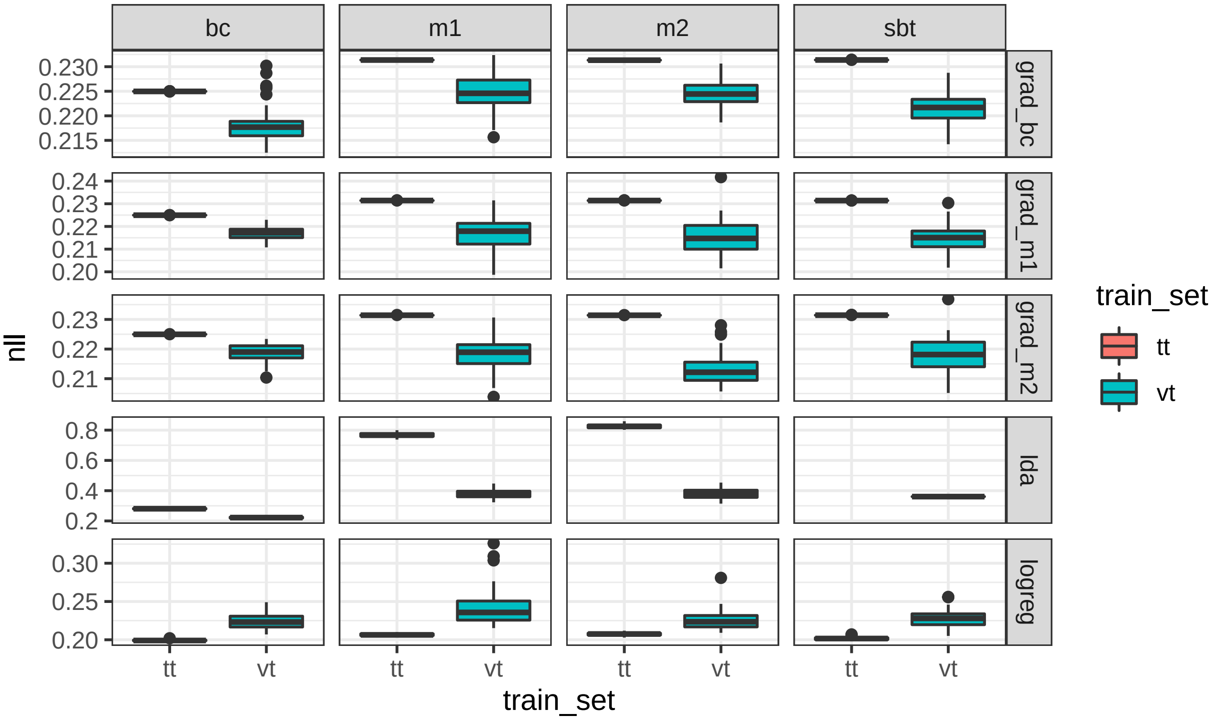


CIFAR−10. Metric ECE of ensembles with combining method trained on different train sets
networks clip_ViT_B_32_LP xception



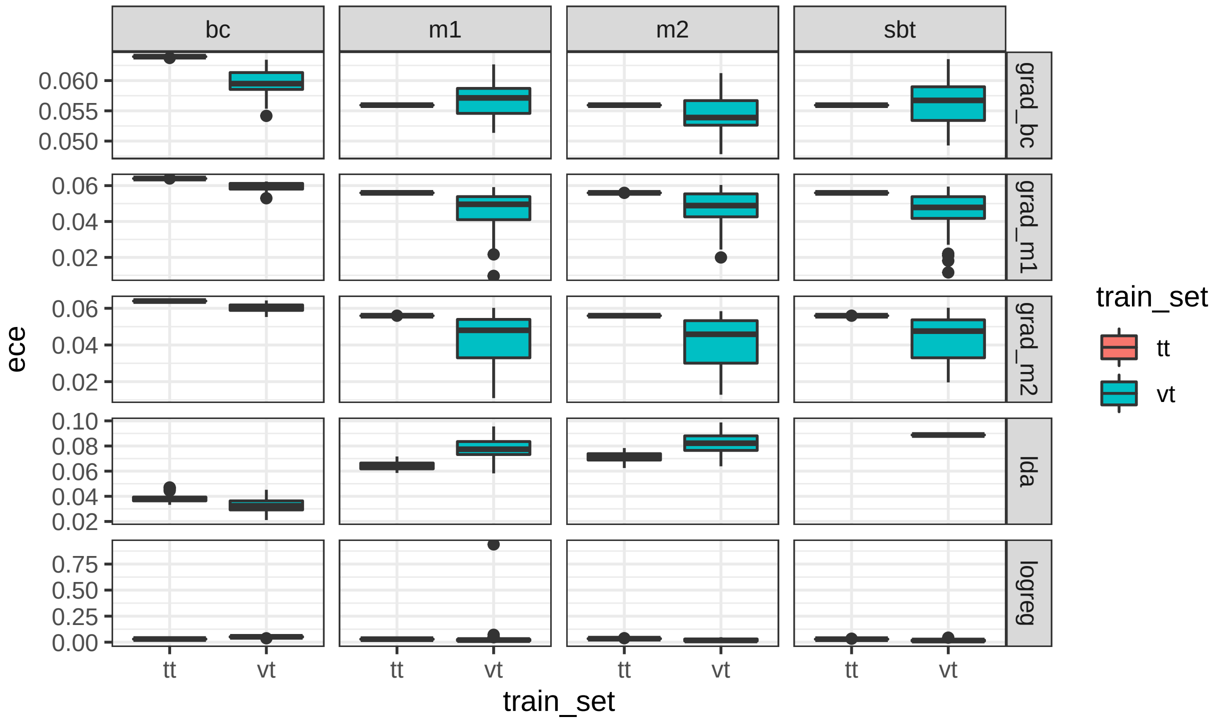
```
## Warning: Removed 99 rows containing non-finite values (stat_boxplot).
```

CIFAR−10. Metric NLL of ensembles with combining method
trained on different train sets
networks densenet121 resnet34

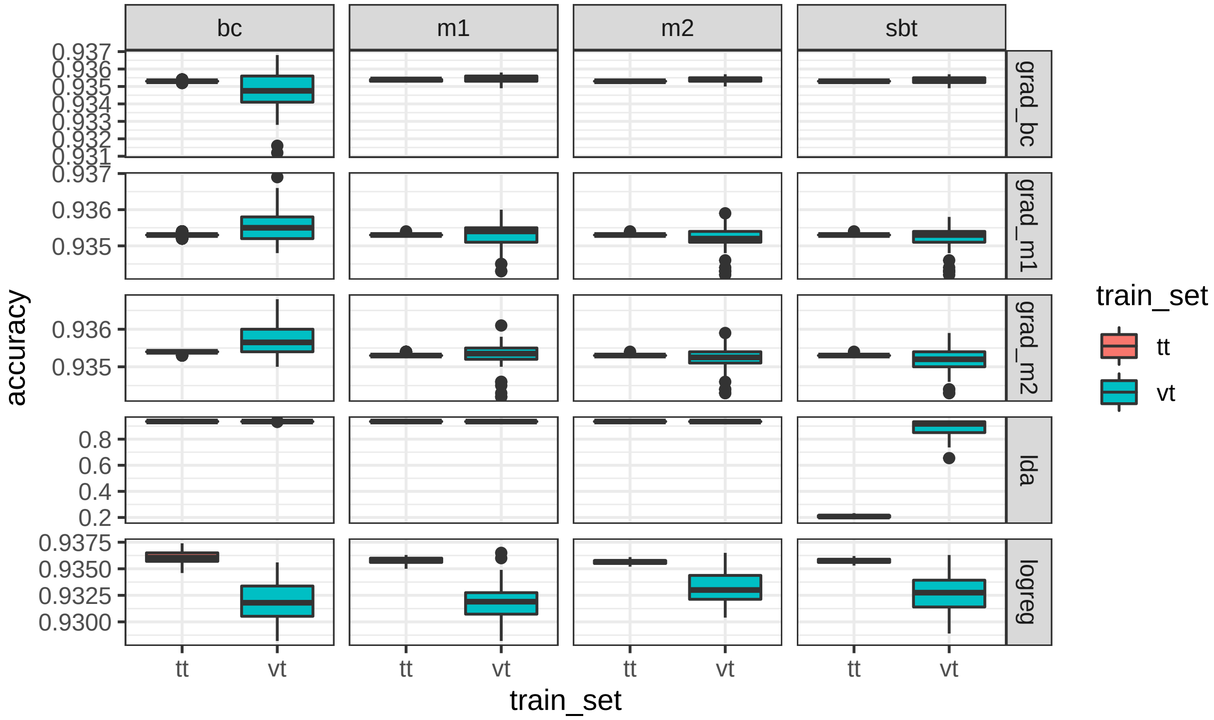


```
## Warning: Removed 99 rows containing non-finite values (stat_boxplot).
```

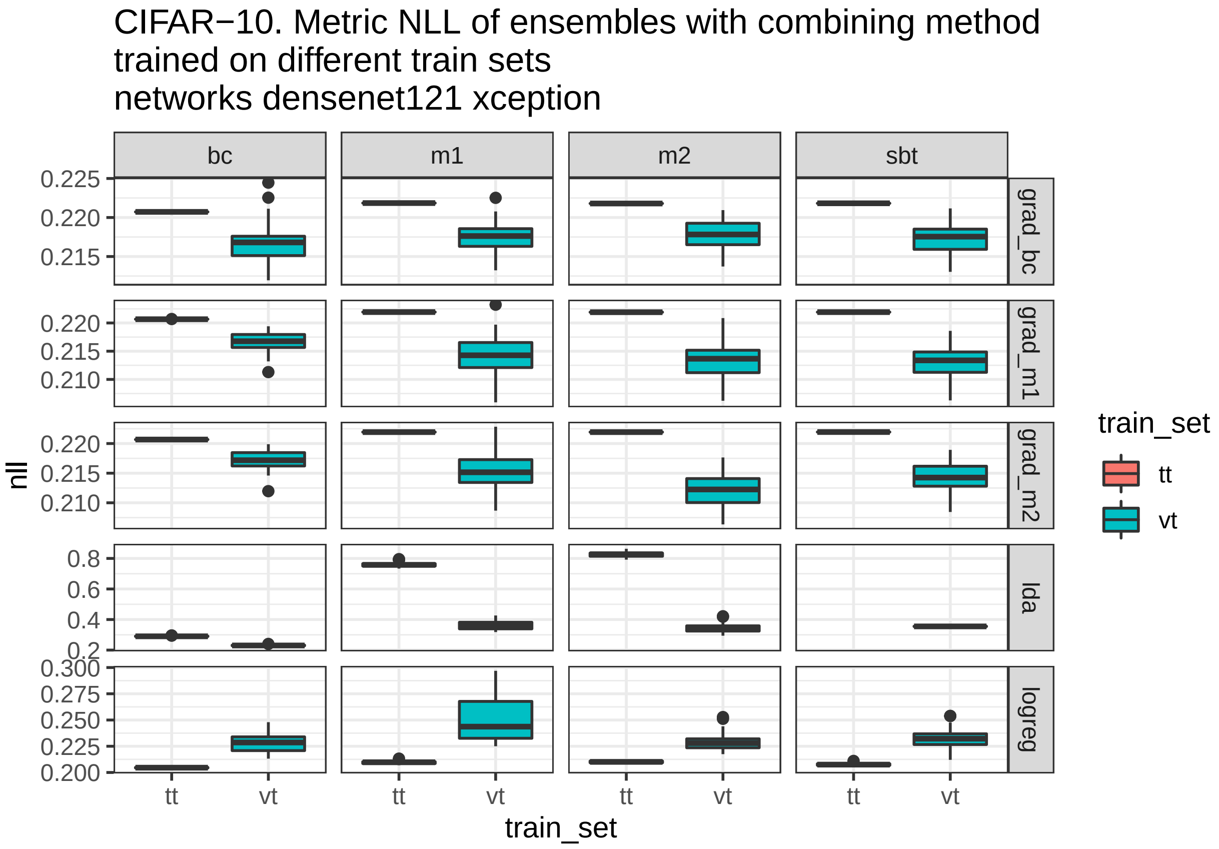
CIFAR−10. Metric ECE of ensembles with combining method trained on different train sets
networks densenet121 resnet34



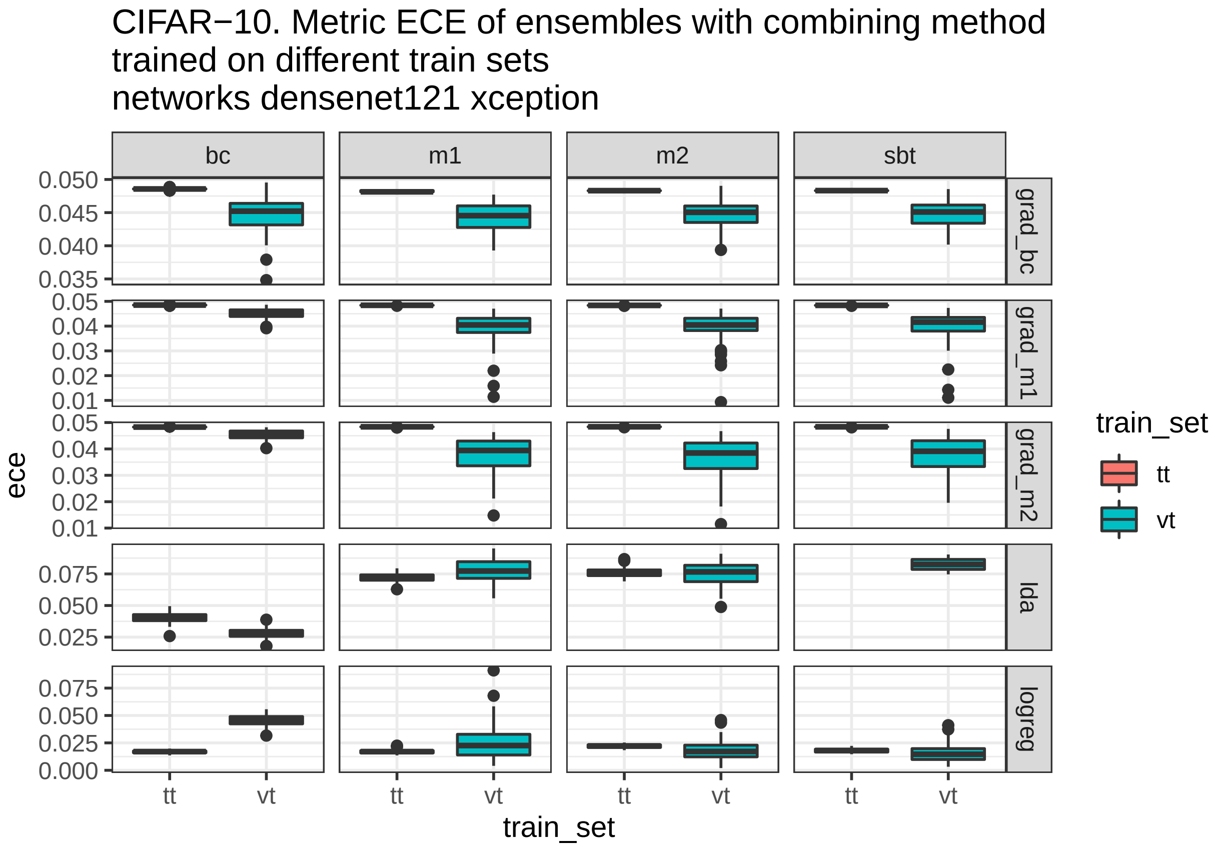
CIFAR−10. Metric accuracy of ensembles with combining method
trained on different train sets
networks densenet121 xception

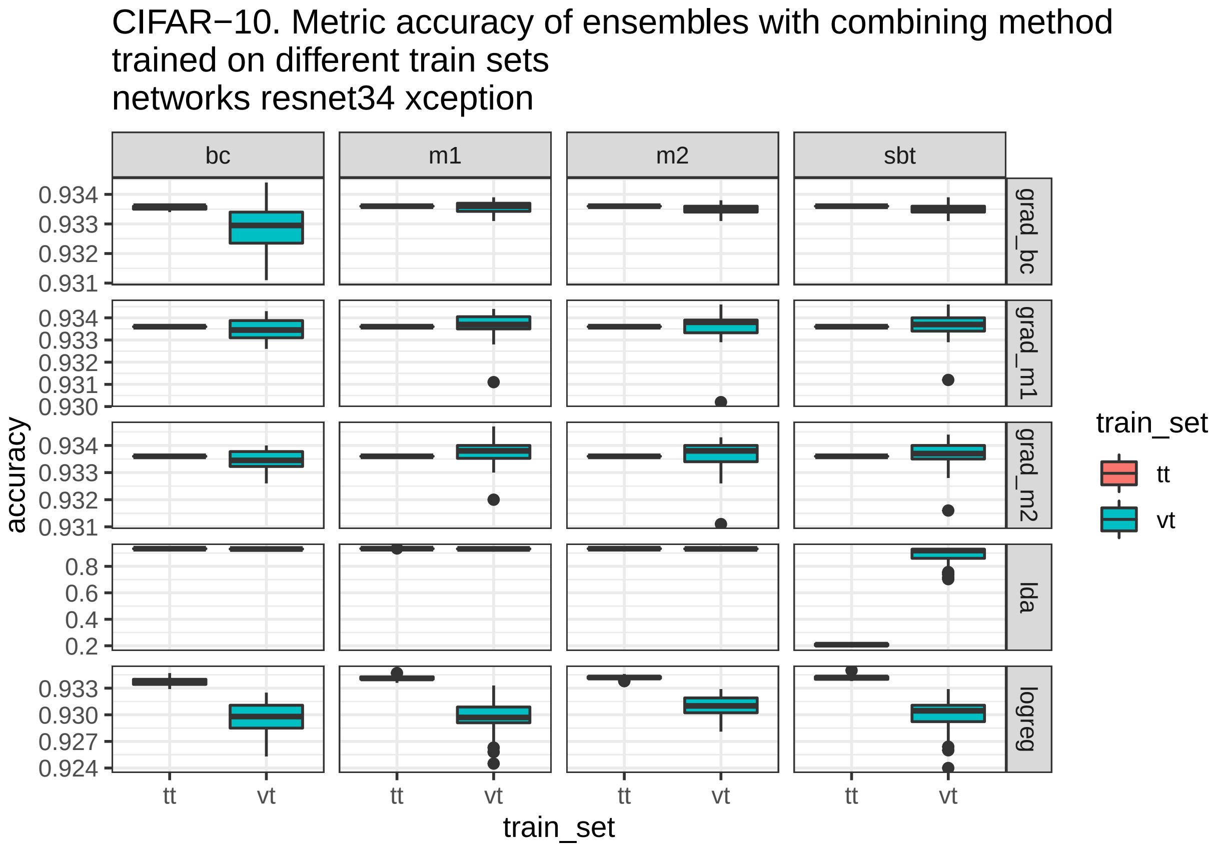


```
## Warning: Removed 98 rows containing non-finite values (stat_boxplot).
```

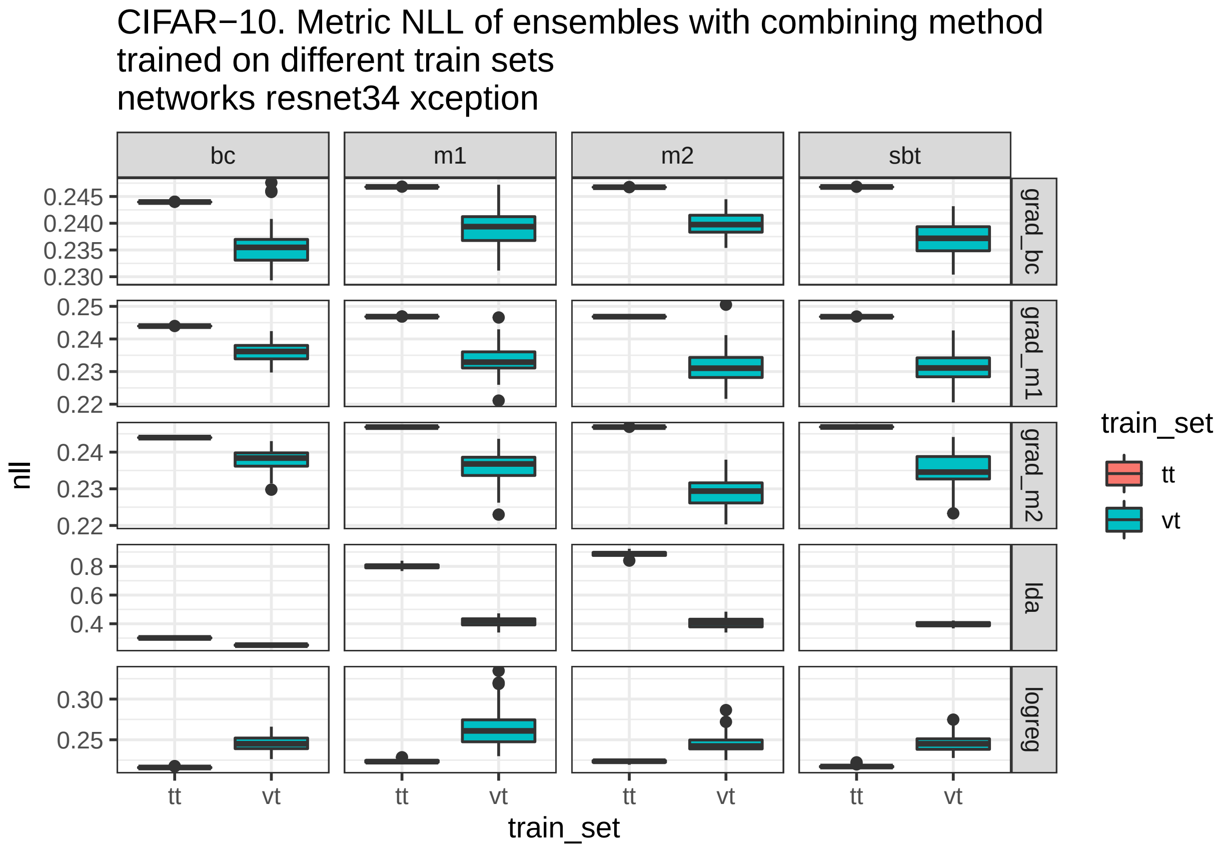


```
## Warning: Removed 98 rows containing non-finite values (stat_boxplot).
```

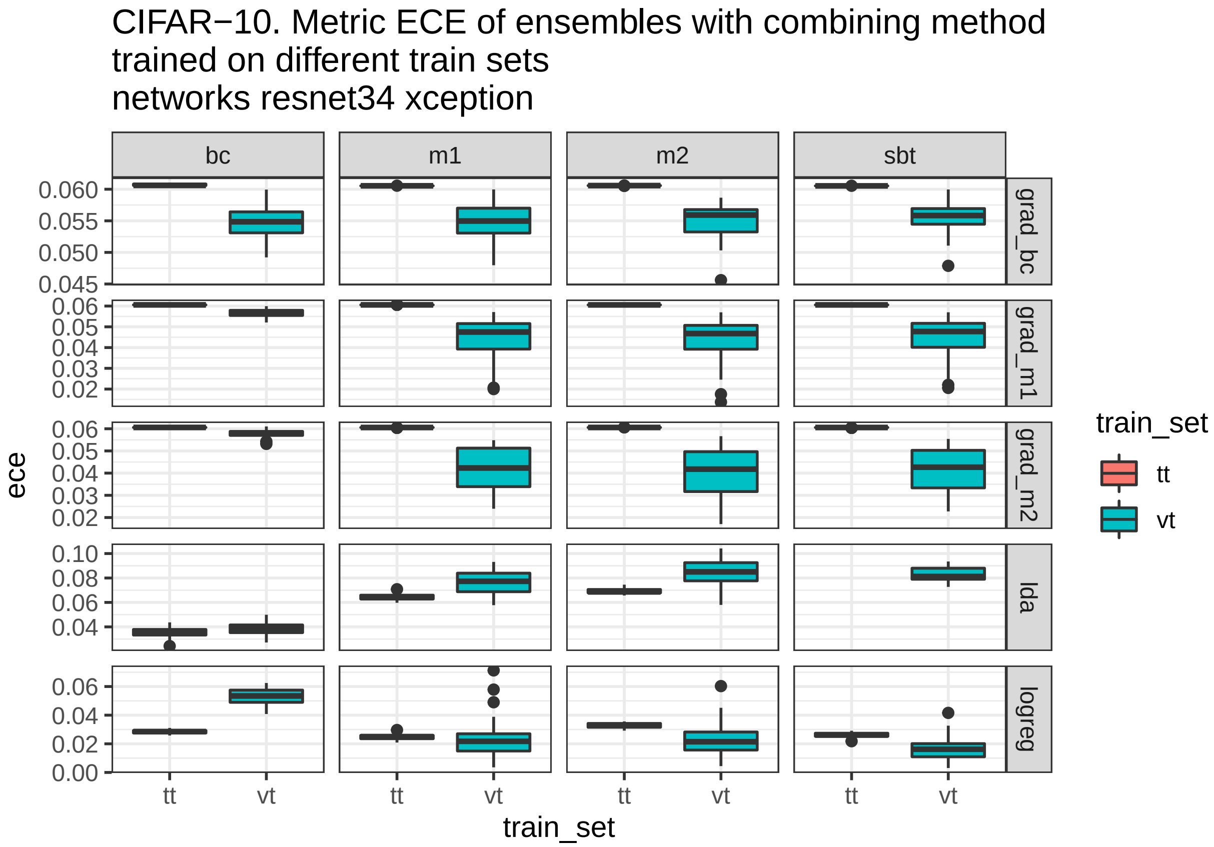

CIFAR−10. Metric ECE of ensembles with combining method
trained on different train sets
networks densenet121 xception
bc
m1
m2
sbt
grad_bc
grad_m1
grad_m2
lda
logreg
0.050
0.045
0.040
0.035
0.05
0.04
0.03
0.02
0.01
0.075
0.050
0.025
0.000
ece
tt
vt
train_set
train_set
tt
vt



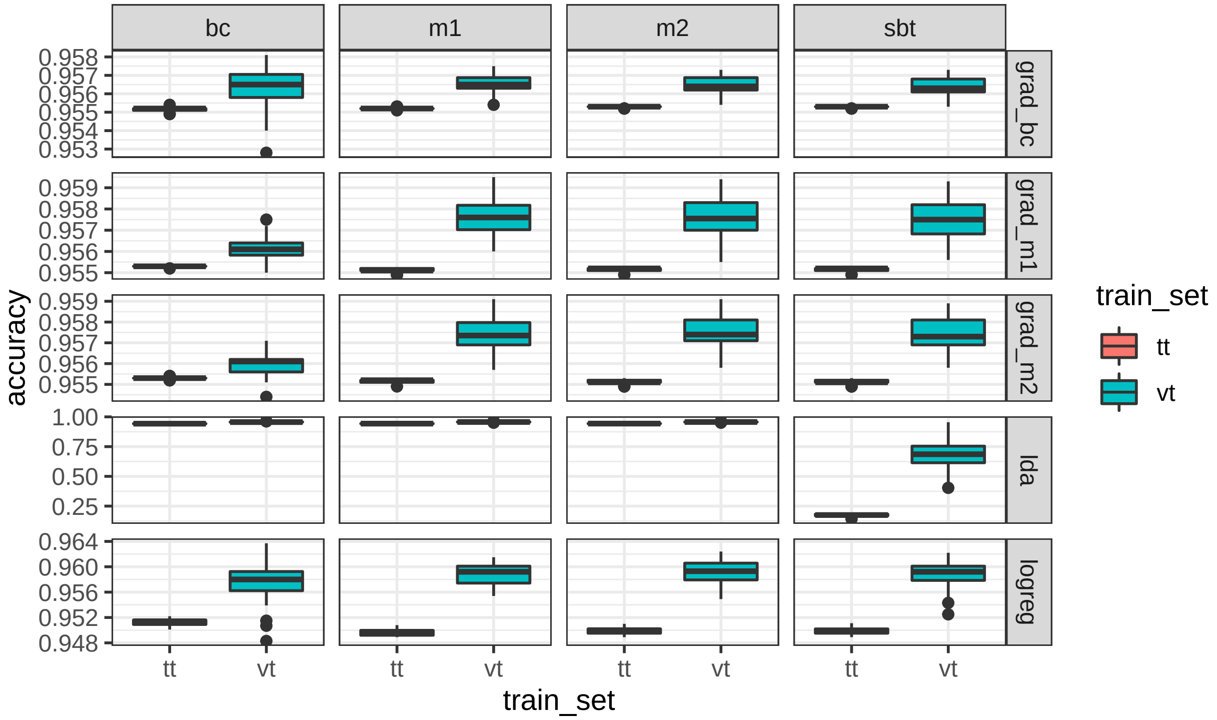
```
## Warning: Removed 94 rows containing non-finite values (stat_boxplot).
```



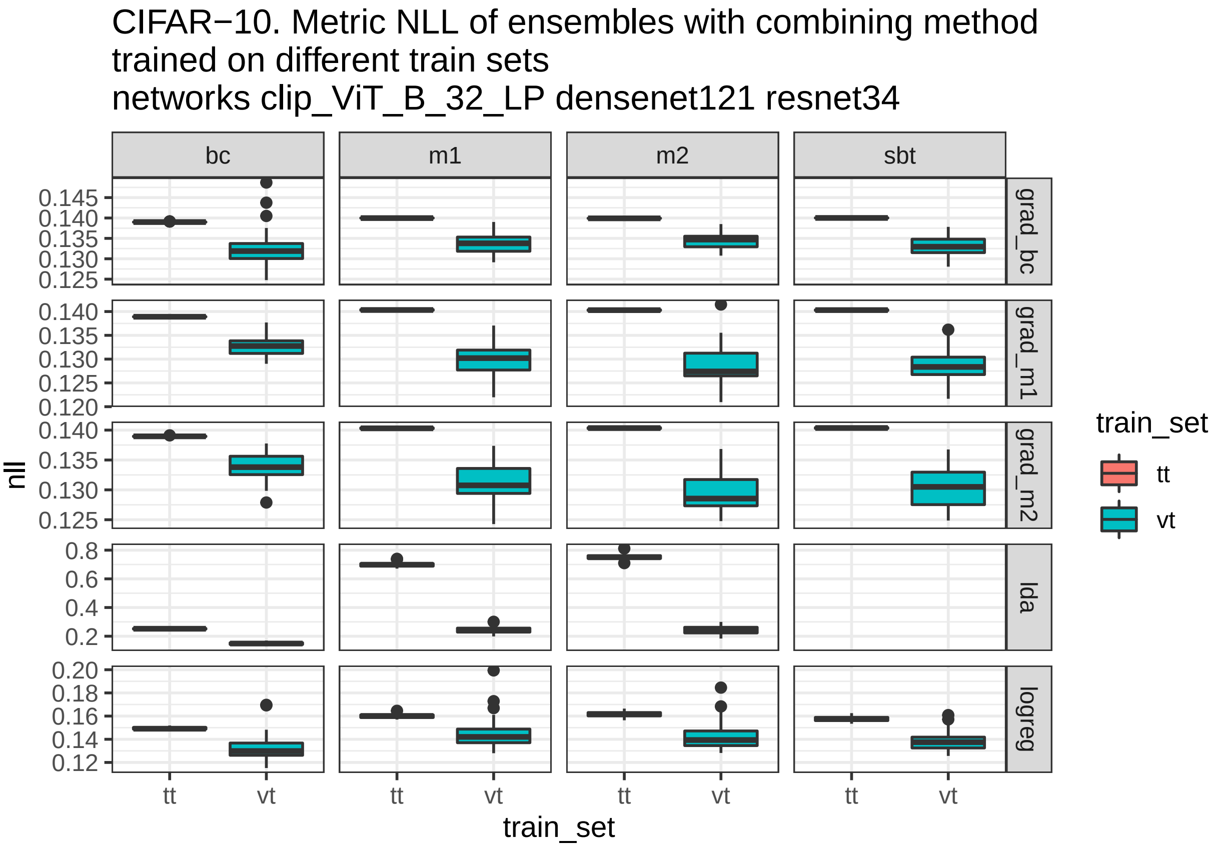
```
## Warning: Removed 94 rows containing non-finite values (stat_boxplot).
```


CIFAR−10. Metric ECE of ensembles with combining method
trained on different train sets
networks resnet34 xception
bc
m1
m2
sbt
grad_bc
grad_m1
grad_m2
lda
logreg
ece
0.060
0.055
0.050
0.045
0.06
0.05
0.04
0.03
0.02
0.10
0.08
0.06
0.04
0.02
0.00
tt
vt
train_set
train_set
tt
vt

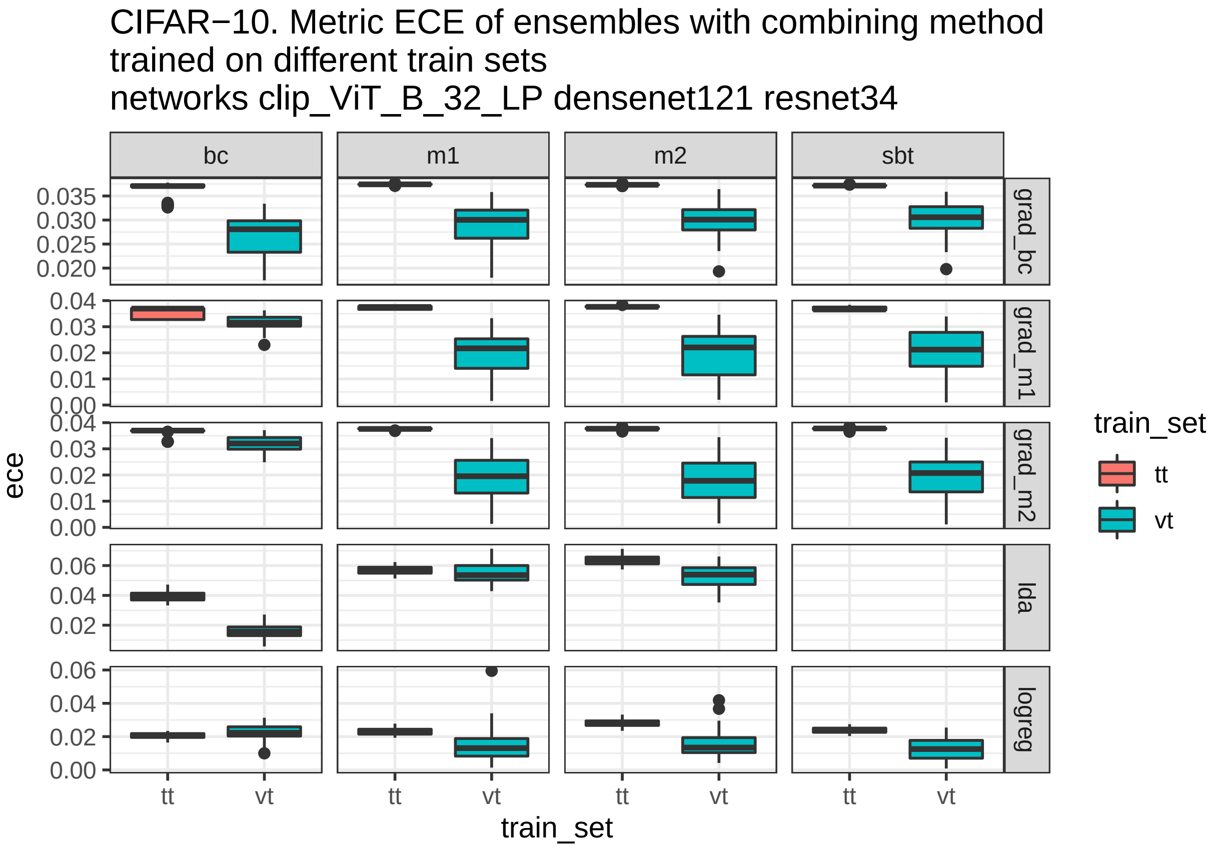
CIFAR−10. Metric accuracy of ensembles with combining method trained on different train sets
networks clip_ViT_B_32_LP densenet121 resnet34



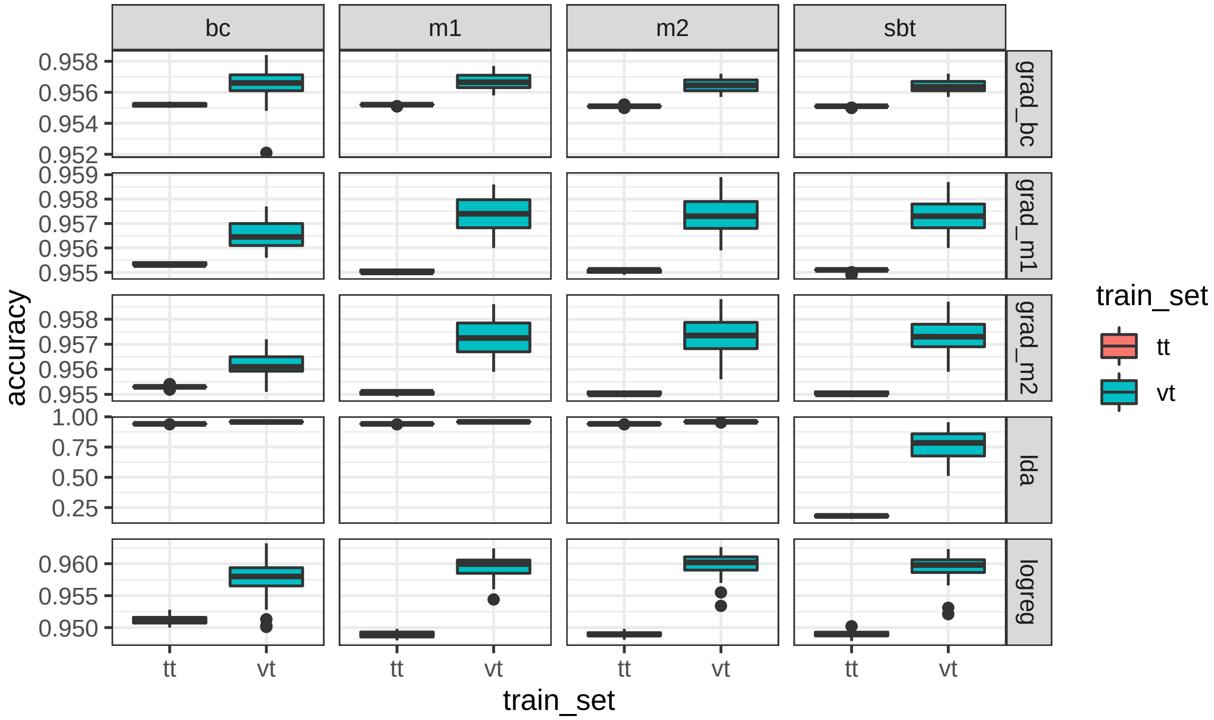
```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```



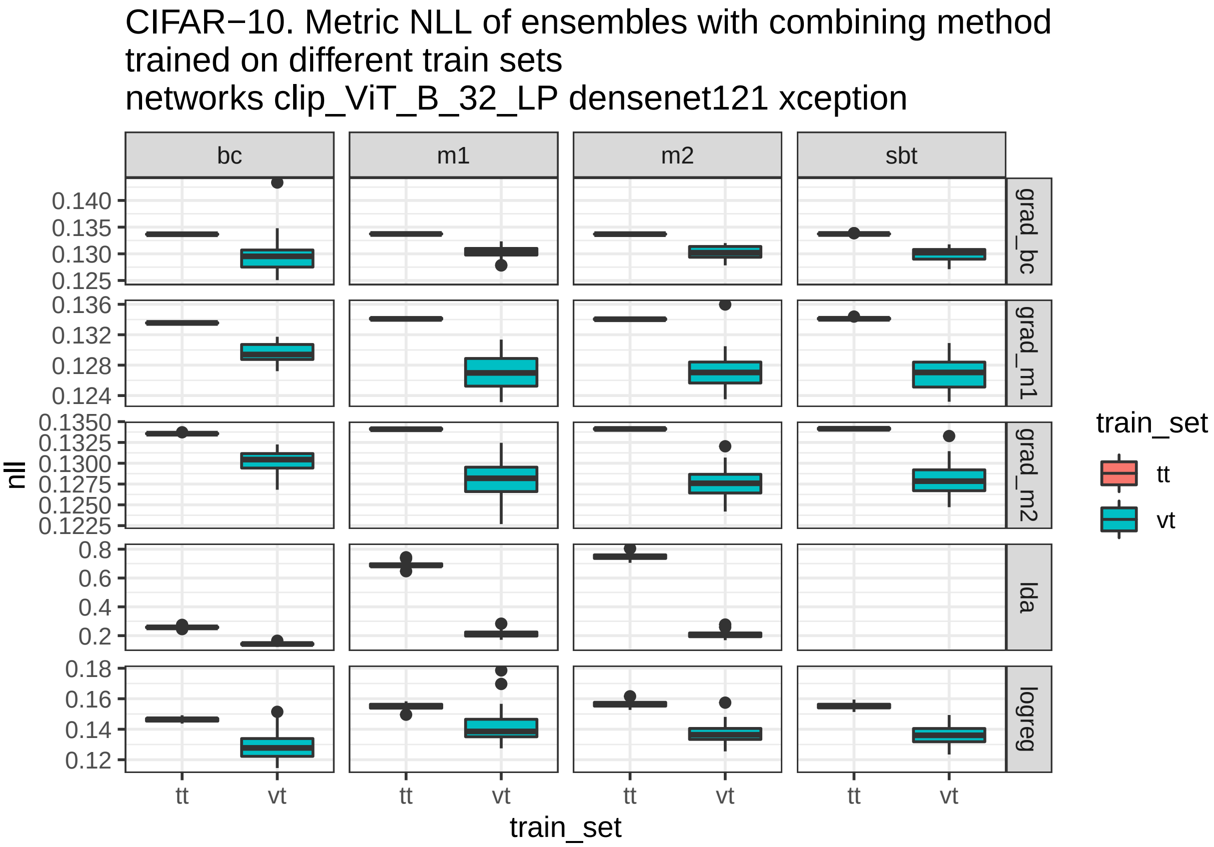
```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```


CIFAR−10. Metric ECE of ensembles with combining method
trained on different train sets
networks clip_ViT_B_32_LP densenet121 resnet34
bc
m1
m2
sbt
grad_bc
grad_m1
grad_m2
lda
logreg
0.035
0.030
0.025
0.020
0.04
0.03
0.02
0.01
0.00
0.06
0.04
0.02
0.00
ece
tt
vt
train_set
train_set
tt
vt

CIFAR−10. Metric accuracy of ensembles with combining method
trained on different train sets
networks clip_ViT_B_32_LP densenet121 xception

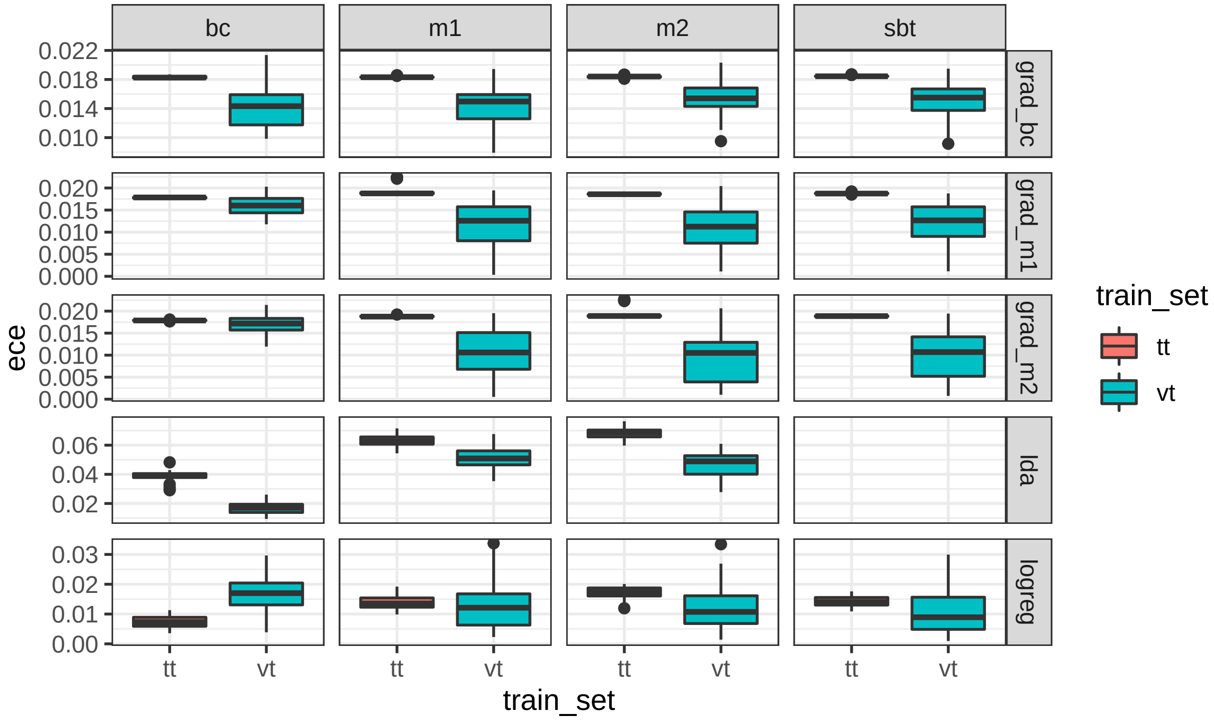


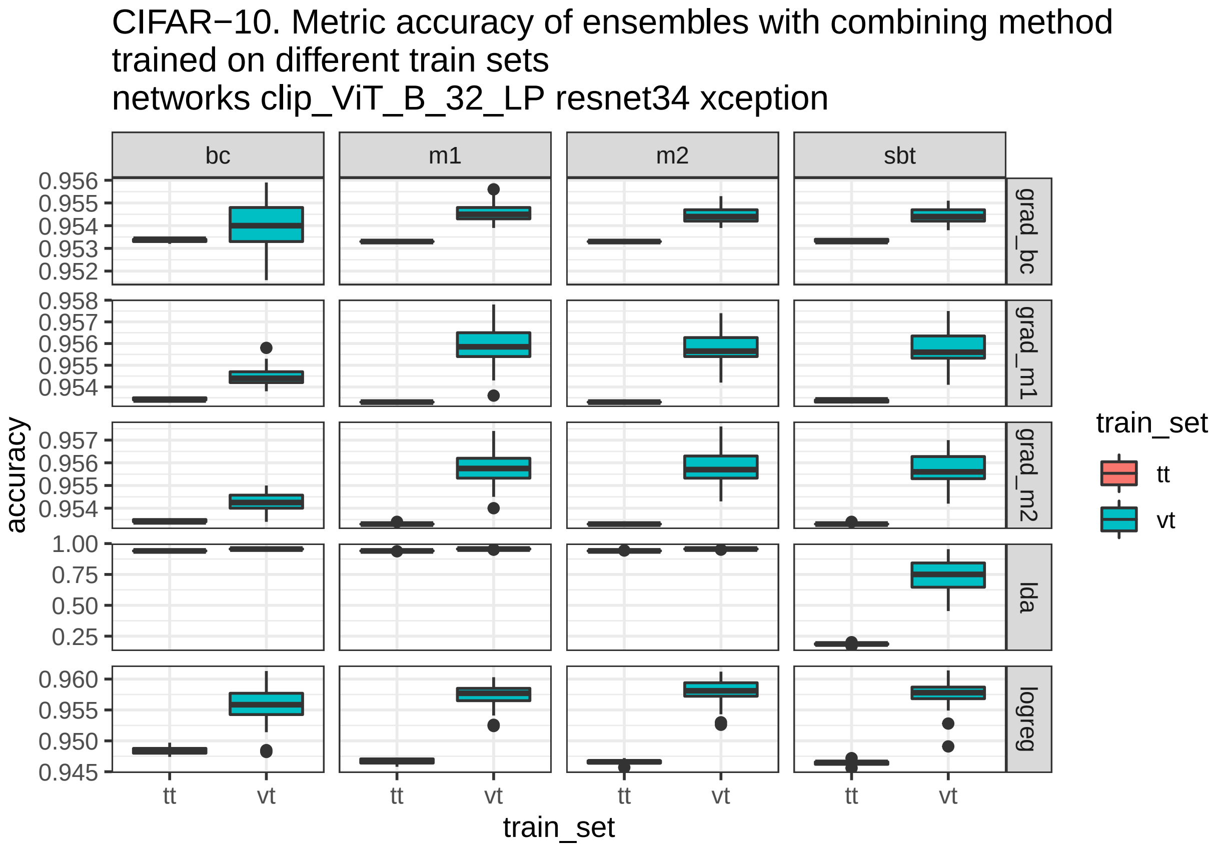
```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```



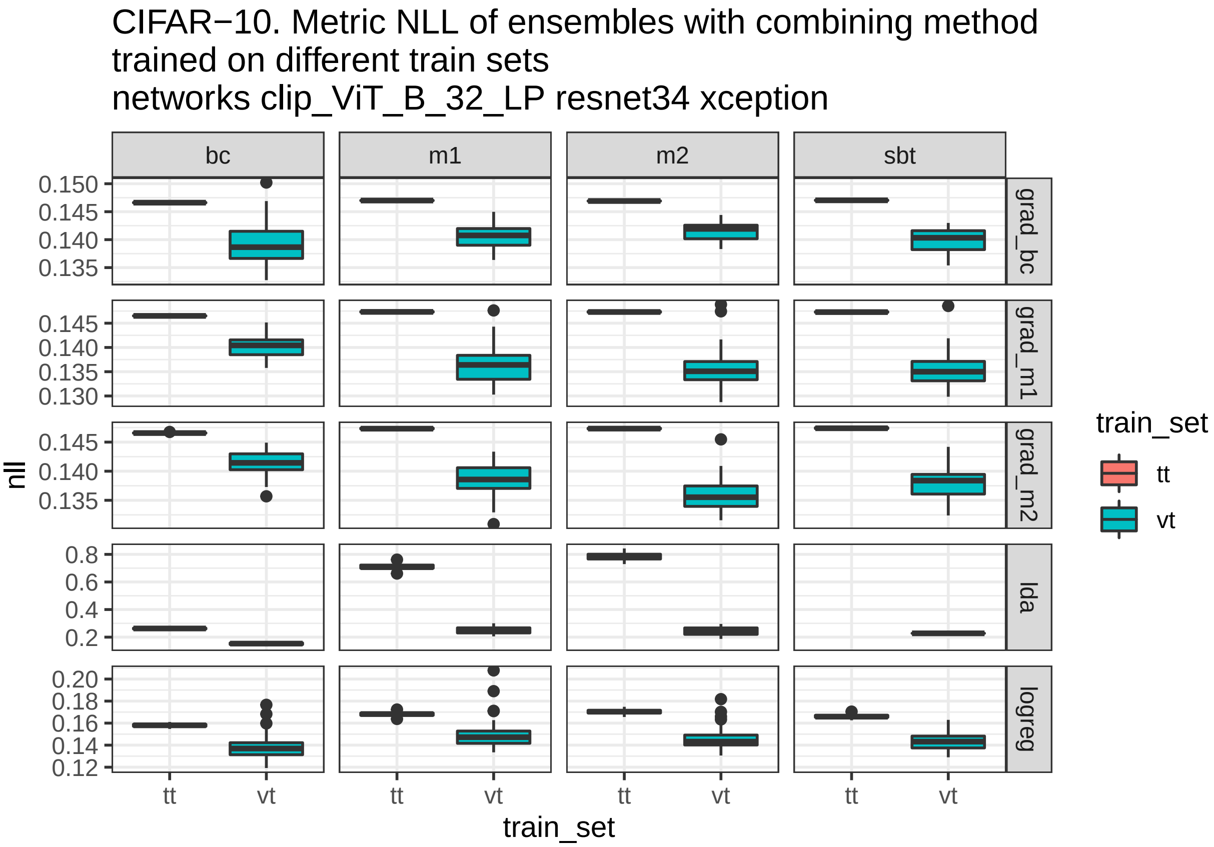
```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```

CIFAR−10. Metric ECE of ensembles with combining method trained on different train sets
networks clip_ViT_B_32_LP densenet121 xception



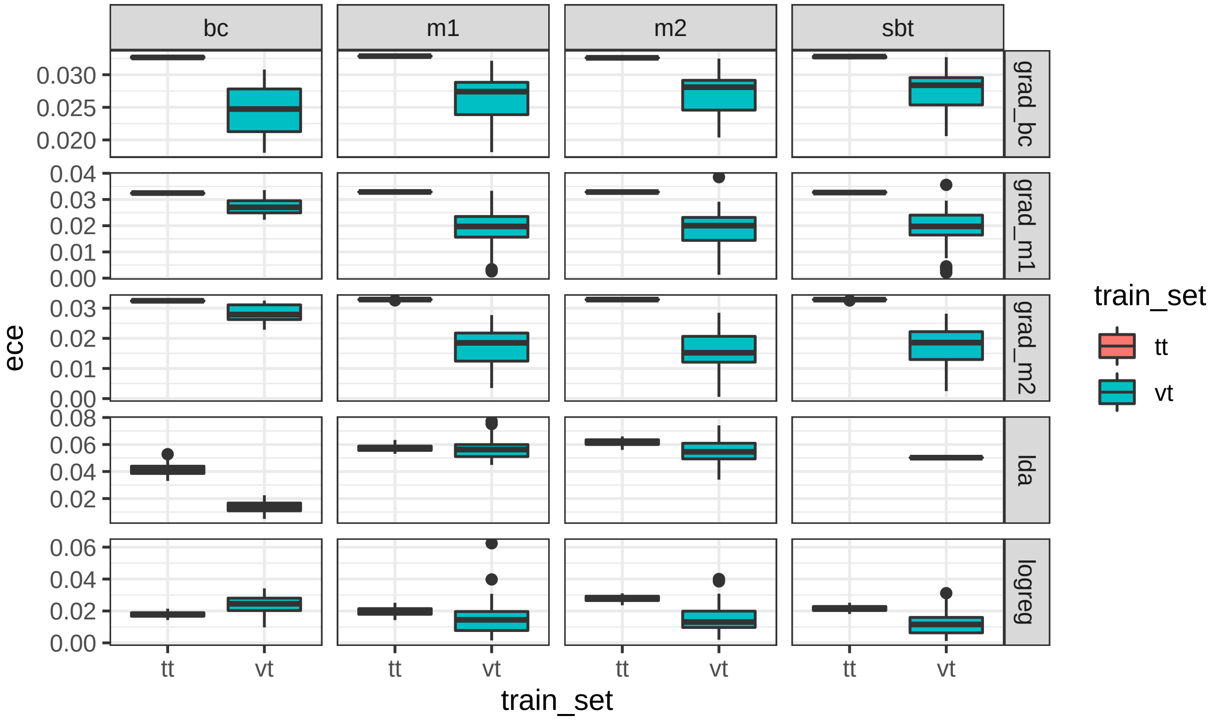


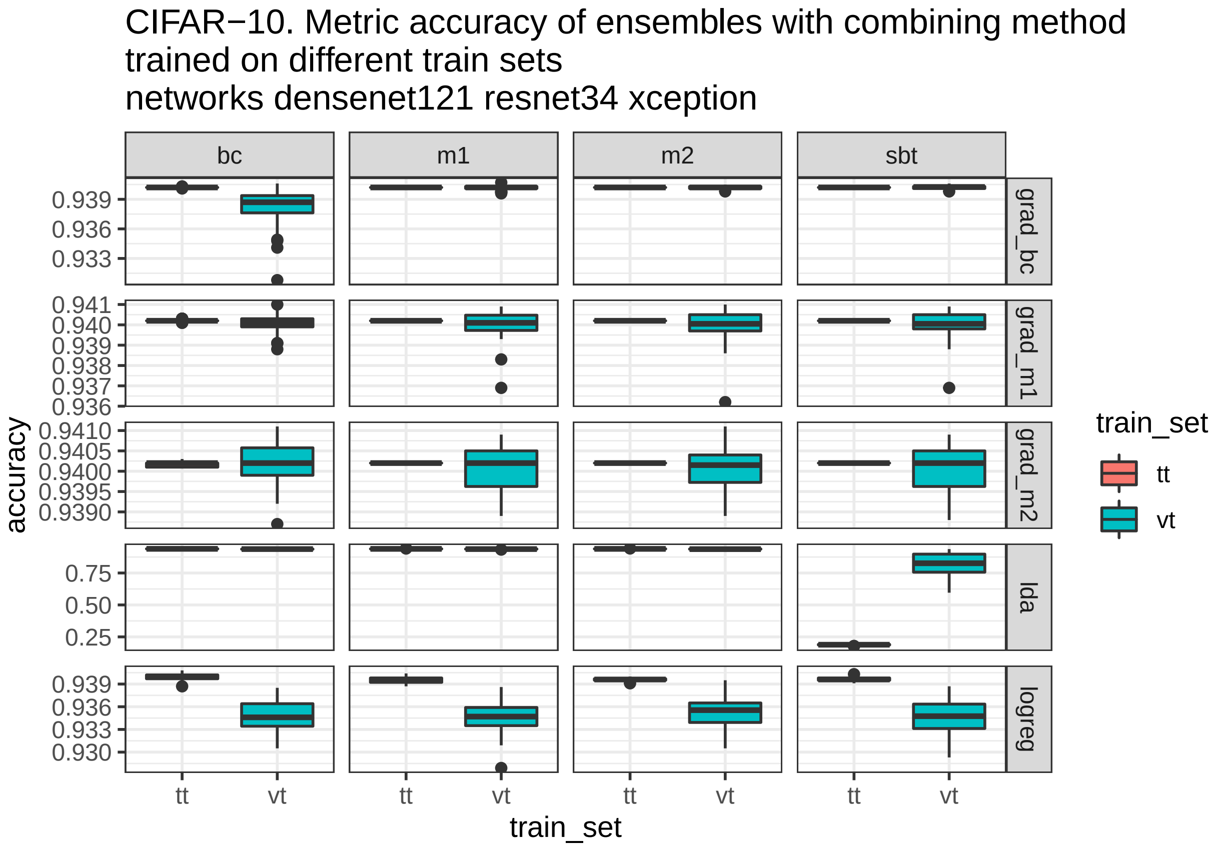
```
## Warning: Removed 99 rows containing non-finite values (stat_boxplot).
```



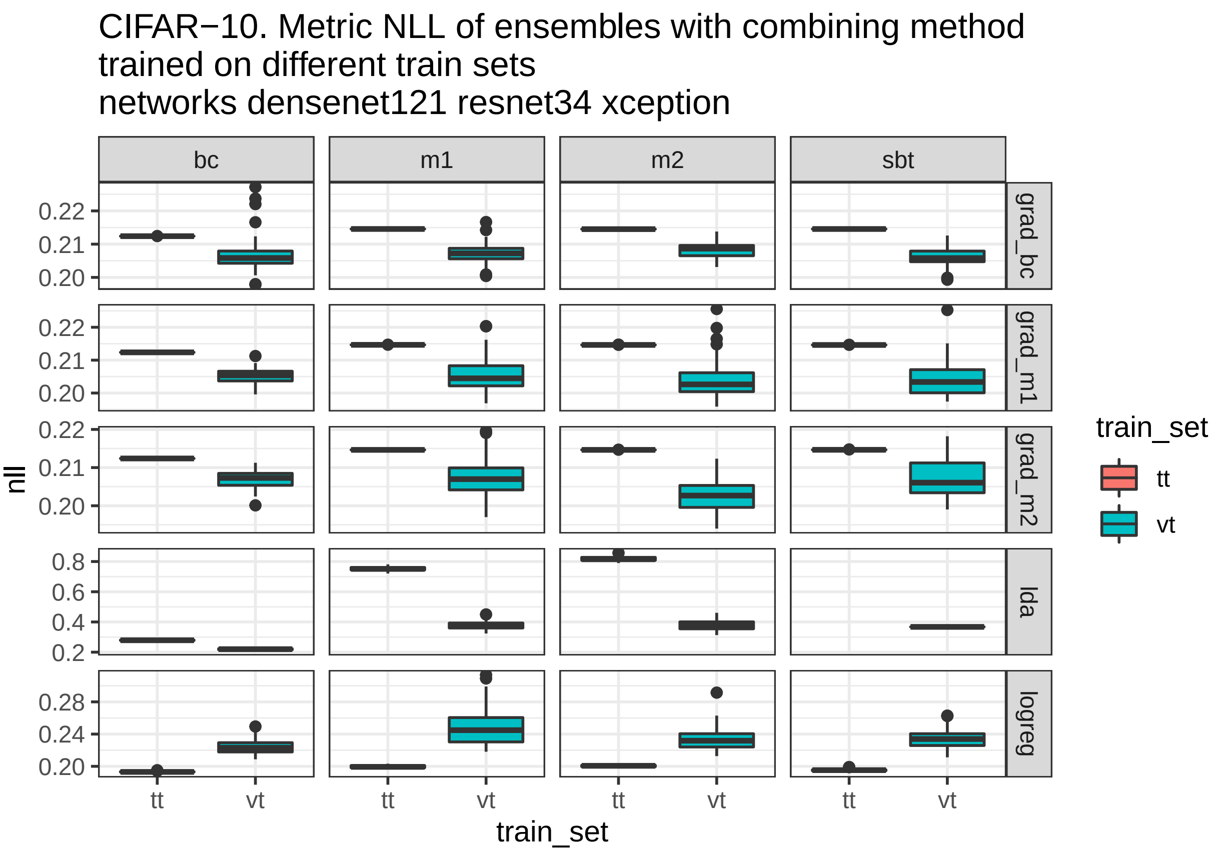
```
## Warning: Removed 99 rows containing non-finite values (stat_boxplot).
```

CIFAR−10. Metric ECE of ensembles with combining method trained on different train sets
networks clip_ViT_B_32_LP resnet34 xception



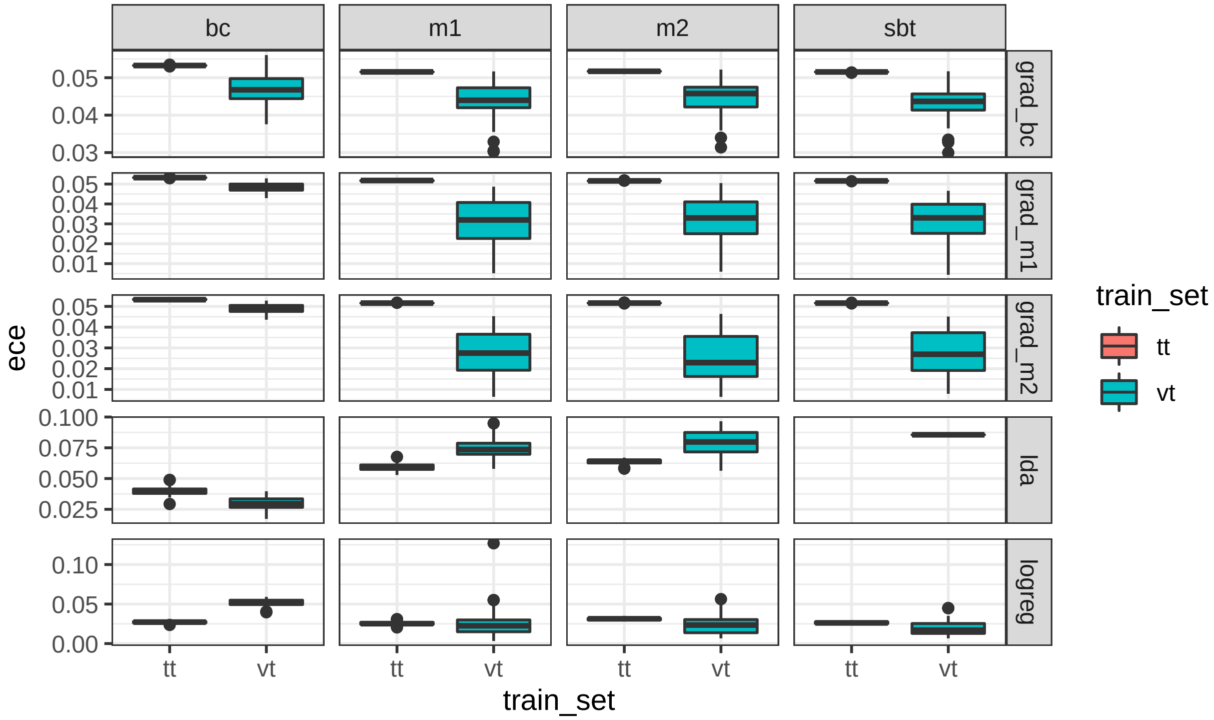


```
## Warning: Removed 99 rows containing non-finite values (stat_boxplot).
```

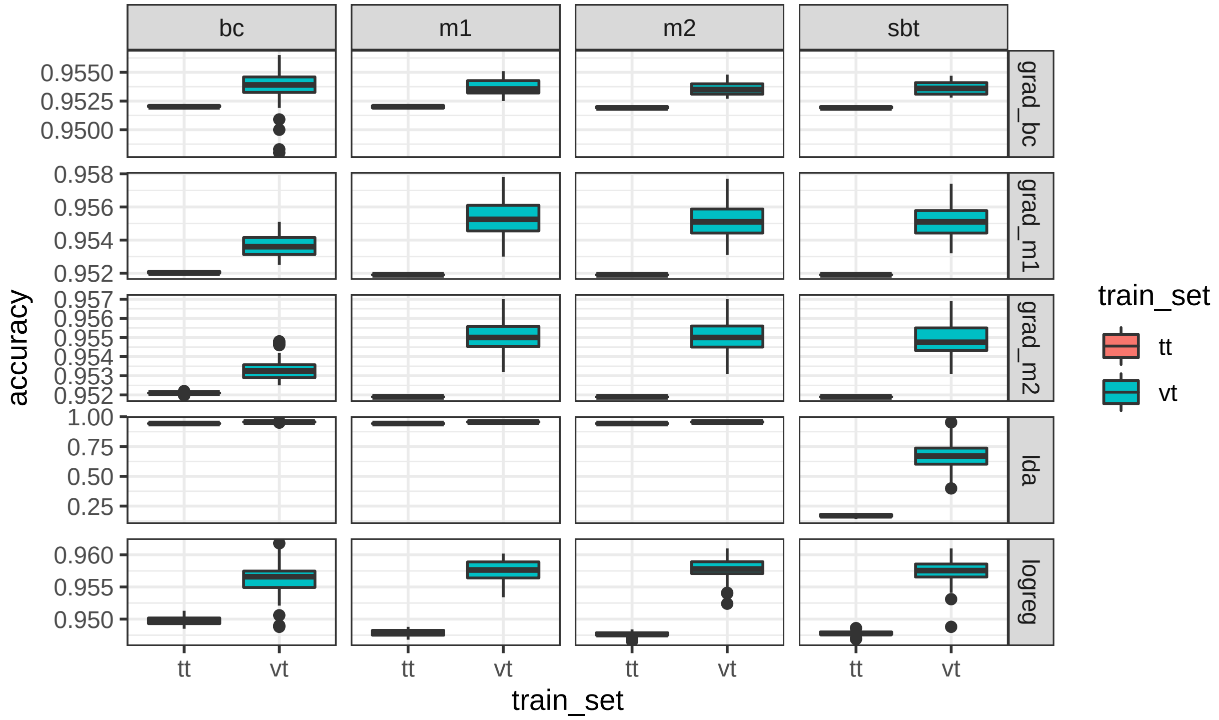


```
## Warning: Removed 99 rows containing non-finite values (stat_boxplot).
```

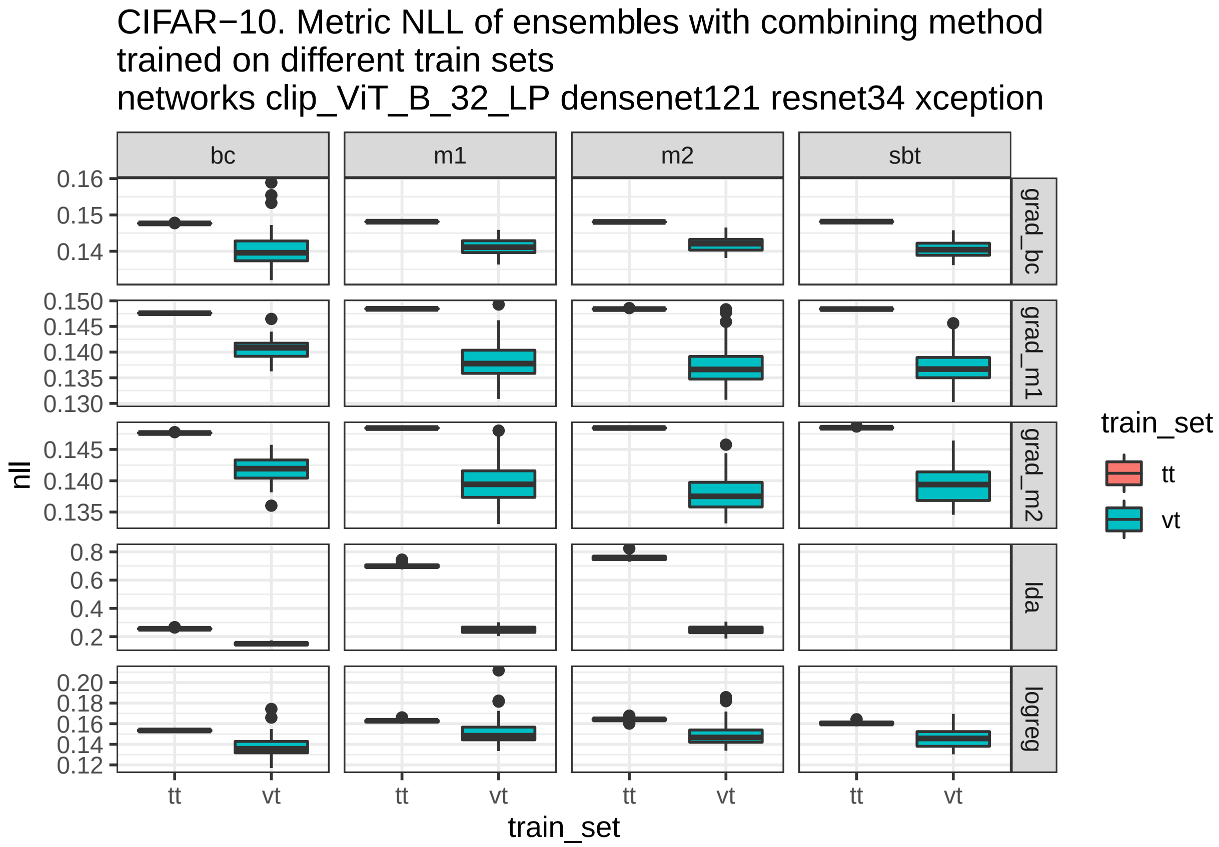
CIFAR−10. Metric ECE of ensembles with combining method trained on different train sets
networks densenet121 resnet34 xception



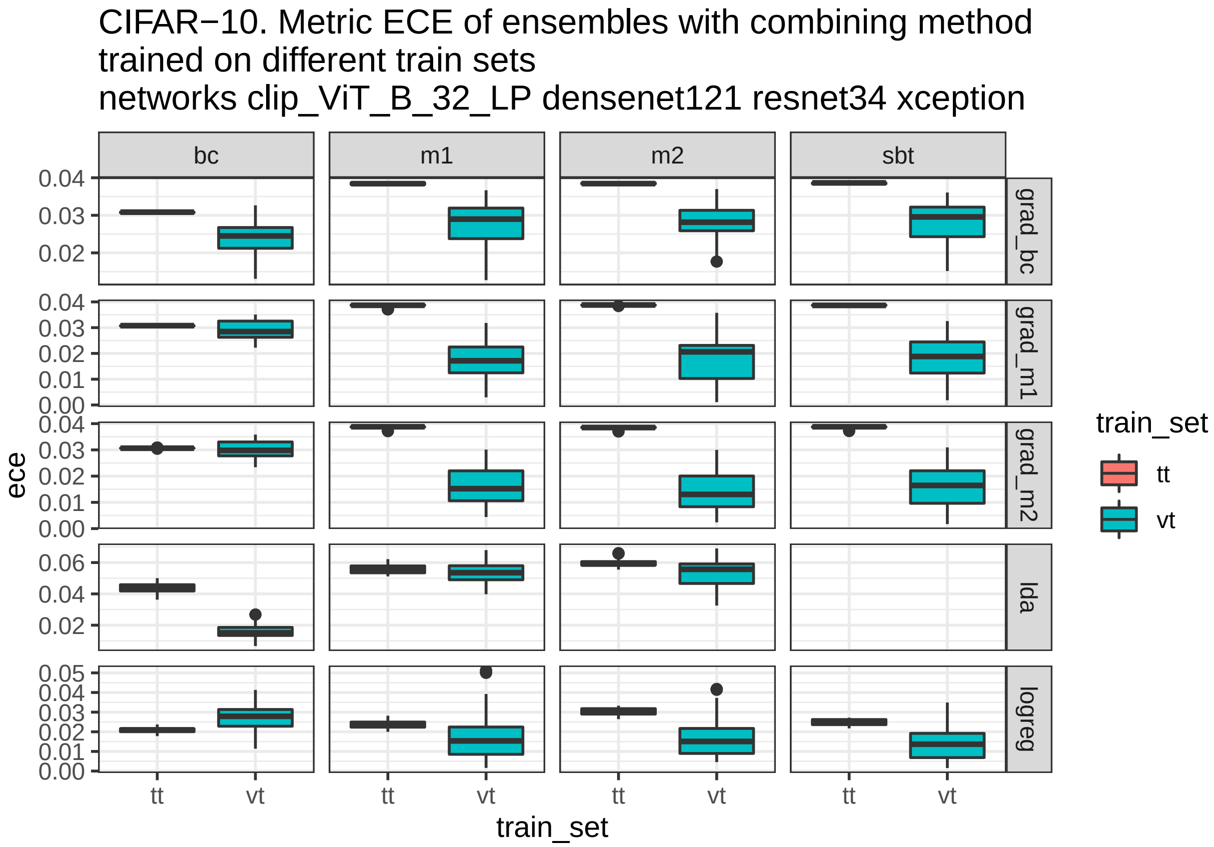
CIFAR−10. Metric accuracy of ensembles with combining method
trained on different train sets
networks clip_ViT_B_32_LP densenet121 resnet34 xception



```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```

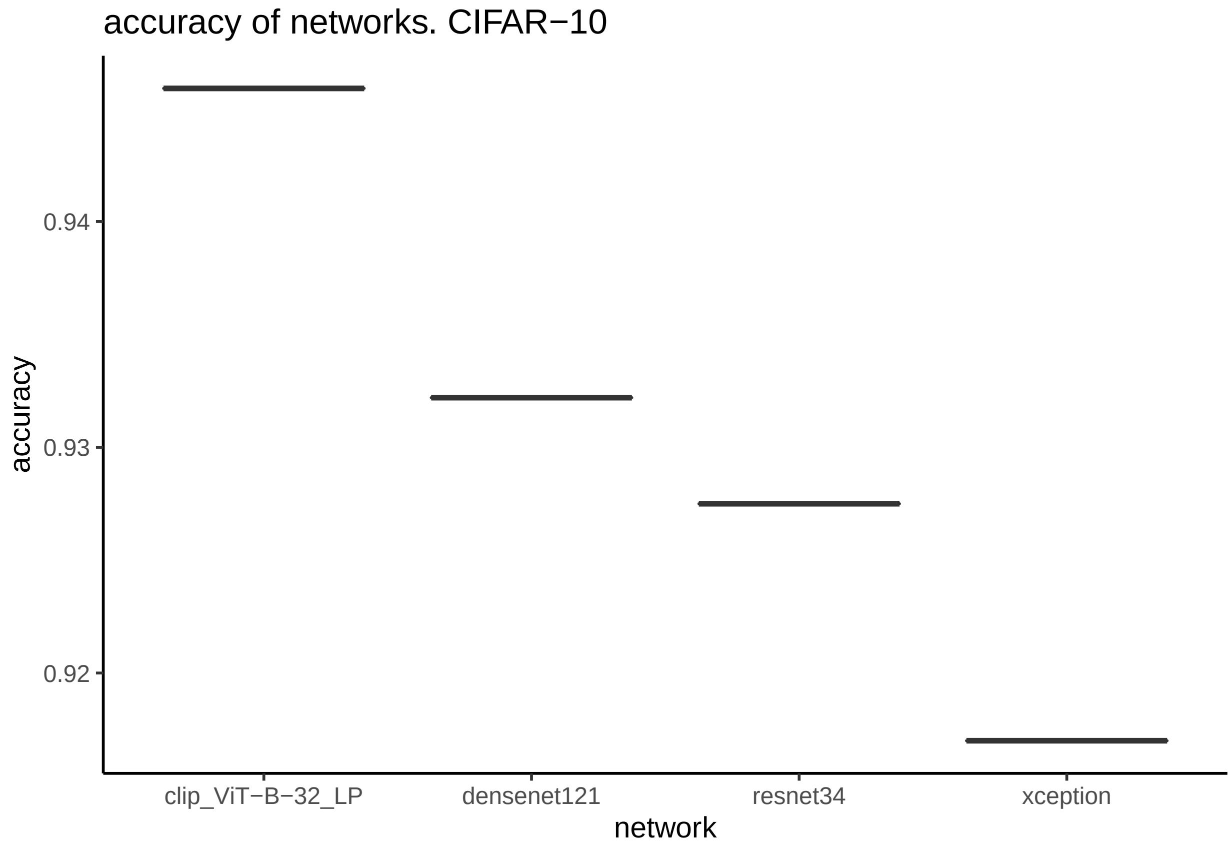


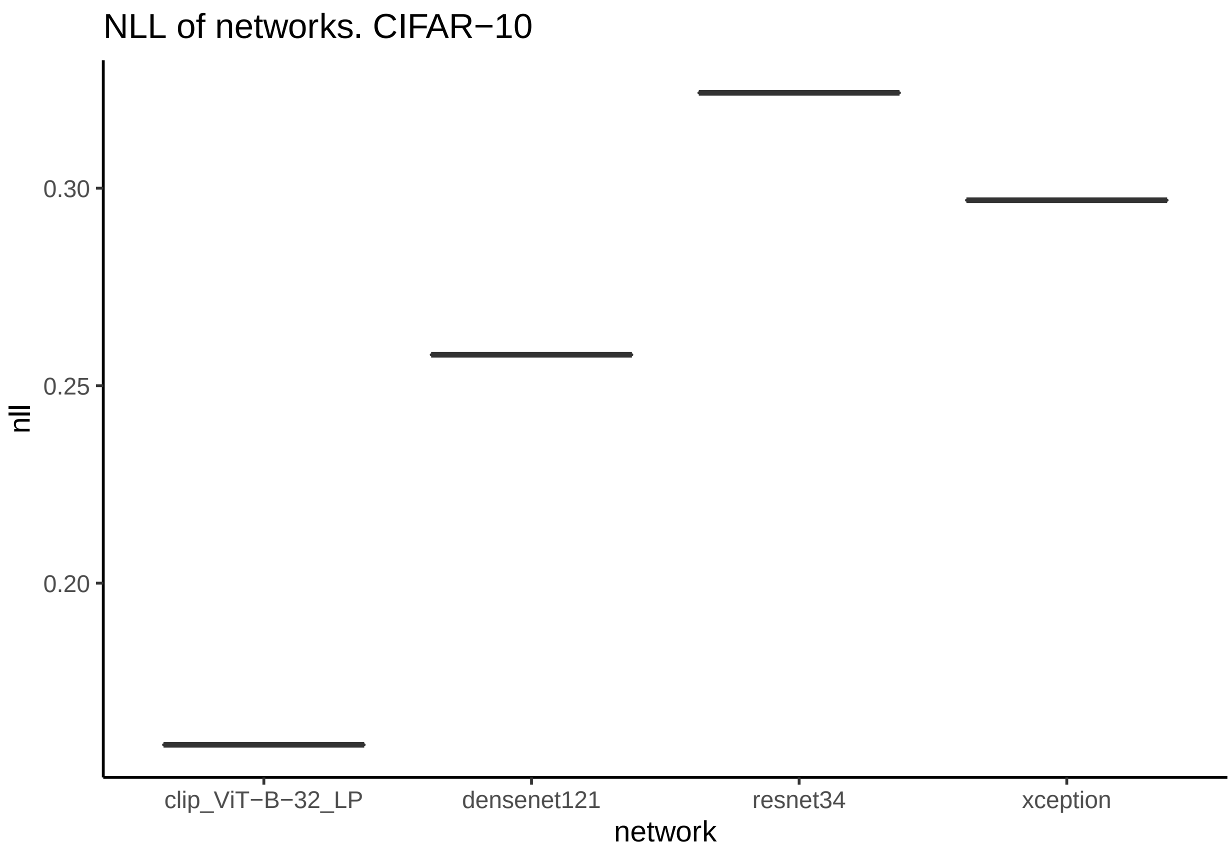
```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```

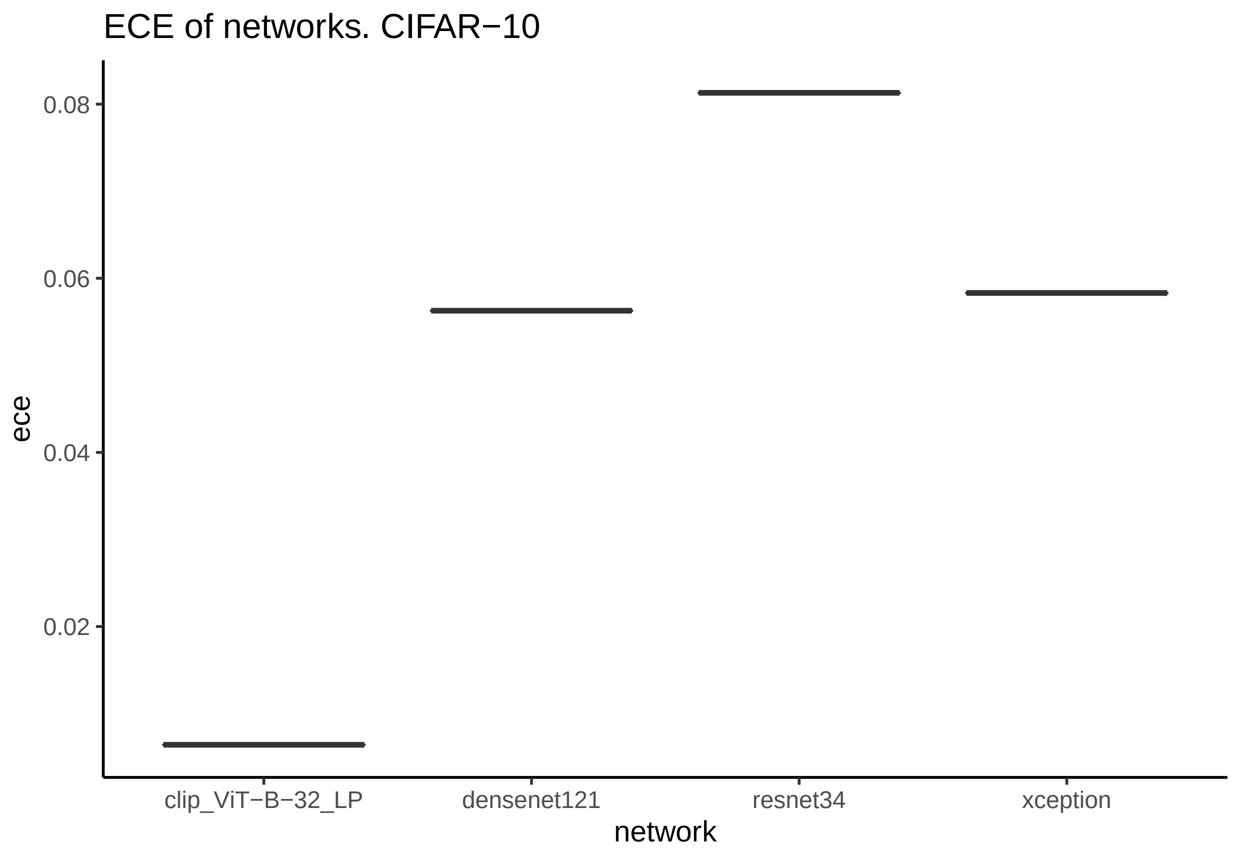


Training on the same training data as the neural networks were trained on seems in majority of cases to provide worse accuracy, worse nll and worse ece compared to training on separate validation set.

```
for (met_i in seq_along(metrics))
{
    box_net <- net_results_c10 %>% ggplot() +
        geom_boxplot(mapping=aes_string(x = "network", y = metrics[met_i])) +
        ggtitle(paste0(metric_names[met_i], " of networks. CIFAR-10")) +
        theme_classic()
    print(box_net)
}
```


accuracy of networks. CIFAR−10
accuracy
0.94
0.93
0.92
clip_ViT−B−32_LP
densenet121
resnet34
xception
network

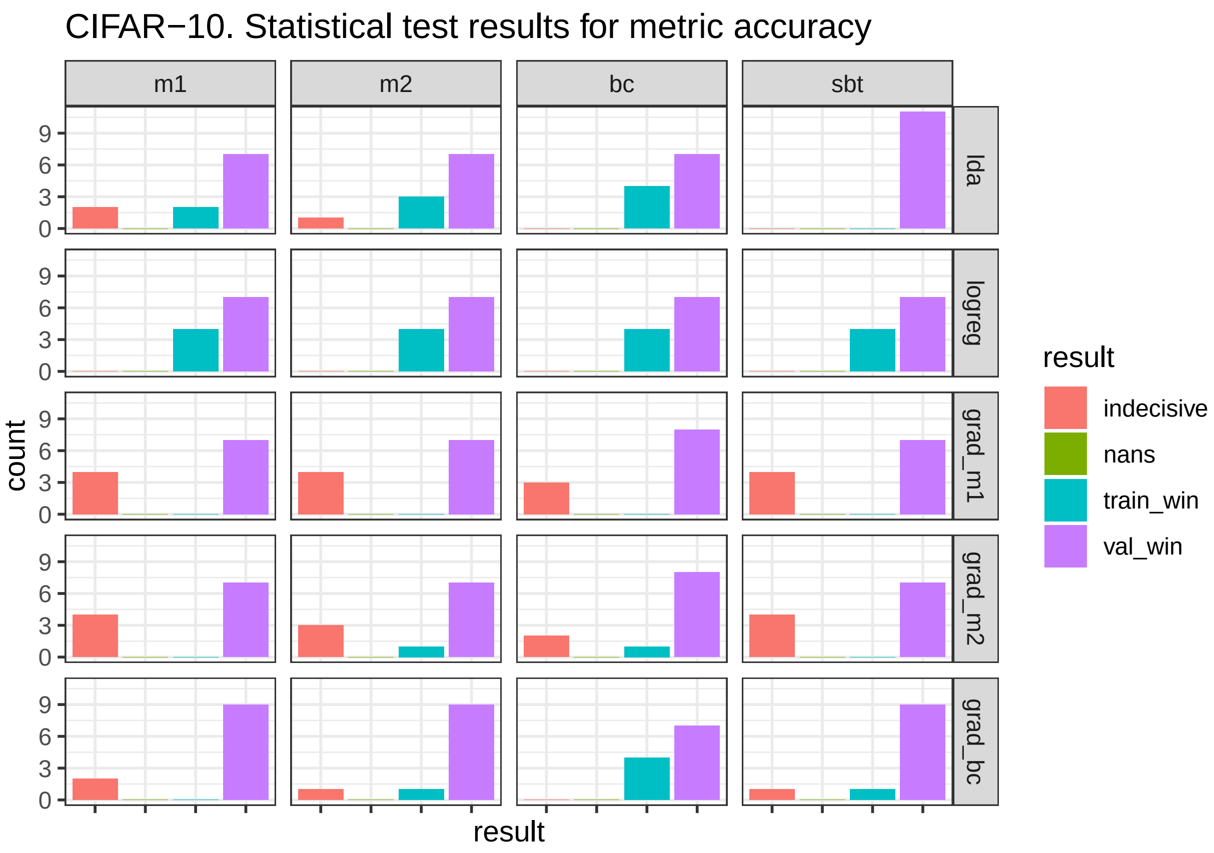

NLL of networks. CIFAR−10
nll
0.30
0.25
0.20
clip_ViT−B−32_LP
densenet121
resnet34
xception
network

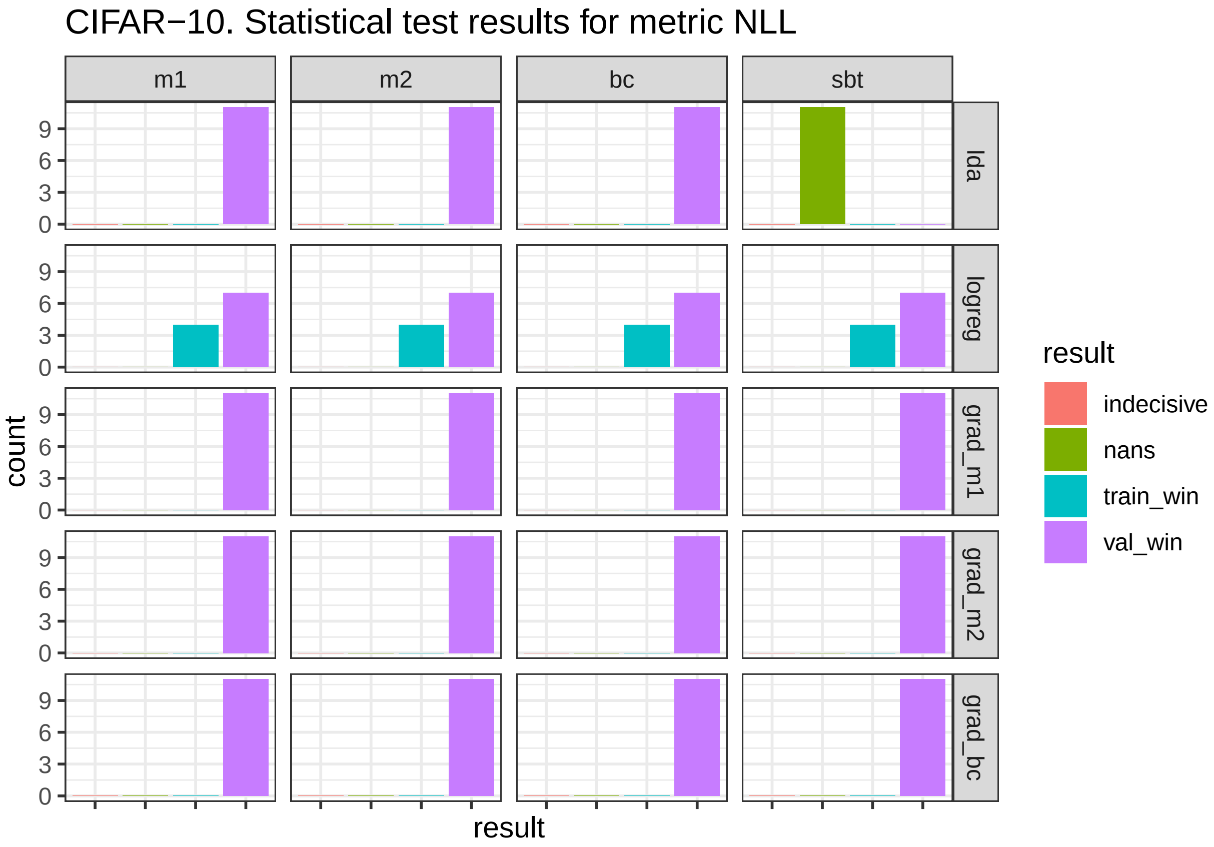


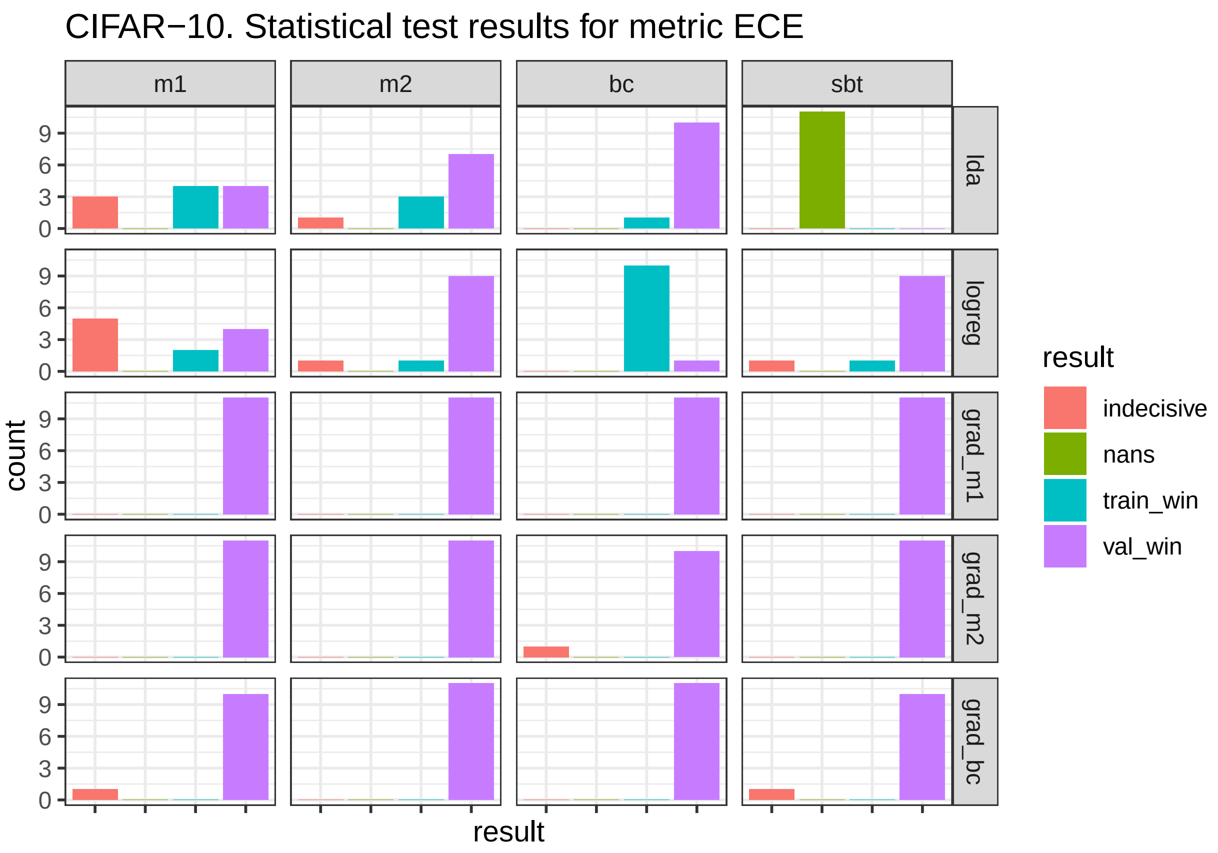
Testing statistical significance of difference for training on validation and train data. We have 50 samples from each distribution, so we will suppose the distributions are normal.

```
tests_df = expand.grid(
    combining_method = unique(ens_results_c10$combining_method),
    coupling_method = unique(ens_results_c10$coupling_method),
    metric = metrics,
    val_win = c(0),
    train_win = c(0),
    indecisive = c(0),
    nans = c(0))
sig_l <- 0.01

for (co_m in unique(ens_results_c10$combining_method))
{
    cur_co_m <- ens_results_c10 %>% filter(combining_method == co_m)
    for (met_i in seq_along(metrics))
    {
        for (cp_m in unique(cur_co_m$coupling_method))
        {
            test_res <- list(val_win = 0, train_win = 0, indecisive = 0, nans = 0)
            for (comb_id in unique(cur_co_m$combination_id))
            {
                cur_co_m_cp_m <- cur_co_m %>% filter(coupling_method == cp_m, combination_id == comb_id
                cur_co_m_cp_m_train <- cur_co_m_cp_m %>% filter(train_set == "tt")
                cur_co_m_cp_m_val <- cur_co_m_cp_m %>% filter(train_set == "vt")
                if (any(is.na(cur_co_m_cp_m_train[[metrics[met_i]]])) |
                    any(is.na(cur_co_m_cp_m_val[[metrics[met_i]]])))
                {
                    test_res[["nans"]] <- test_res[["nans"]] + 1
                }
                else
                {
                    testr <- t.test(cur_co_m_cp_m_train[[metrics[met_i]]], cur_co_m_cp_m_val[[metrics[me
                    if (testr$p.value >= sig_l)
                    {
                        test_res[["indecisive"]] <- test_res[["indecisive"]] + 1
                    }
                    else
                    {
                       if (
                           (metrics_opt[met_i] == "min" & testr$estimate[[1]] > testr$estimate[[2]]) |
                           (metrics_opt[met_i] == "max" & testr$estimate[[1]] < testr$estimate[[2]]))
                        {
                            test_res[["val_win"]] <- test_res[["val_win"]] + 1
                        }
                        else
                        {
                           test_res[["train_win"]] <- test_res[["train_win"]] + 1
                        }
                    }
                }
            }
            tests_df[
                which(tests_df$combining_method == co_m & tests_df$coupling_method == cp_m & tests_df$me
                c("val_win", "train_win", "indecisive", "nans")] <- test_res
        }
    }
}
```

```
tests_df_longer <- pivot_longer(data = tests_df, cols = c("val_win", "train_win", "indecisive", "nans")
for (met_i in seq_along(metrics))
{
    col_plot <- tests_df_longer %>% filter(metric == metrics[met_i]) %>% ggplot() +
        geom_col(mapping = aes(x = result, y = count, fill = result)) +
        facet_grid(rows=vars(combining_method), cols=vars(coupling_method)) +
        ggtitle(paste0("CIFAR-10. Statistical test results for metric ", metric_names[met_i])) +
        theme_bw() +
        theme(
            axis.text.x = element_blank())

    print(col_plot)
}
```


CIFAR−10. Statistical test results for metric accuracy
m1
m2
bc
sbt
lda
logreg
grad_m1
grad_m2
grad_bc
count
9
6
3
0
result
result
indecisive
nans
train_win
val_win


CIFAR−10. Statistical test results for metric NLL
m1
m2
bc
sbt
lda
logreg
grad_m1
grad_m2
grad_bc
0
3
6
9
count
result
result
indecisive
nans
train_win
val_win

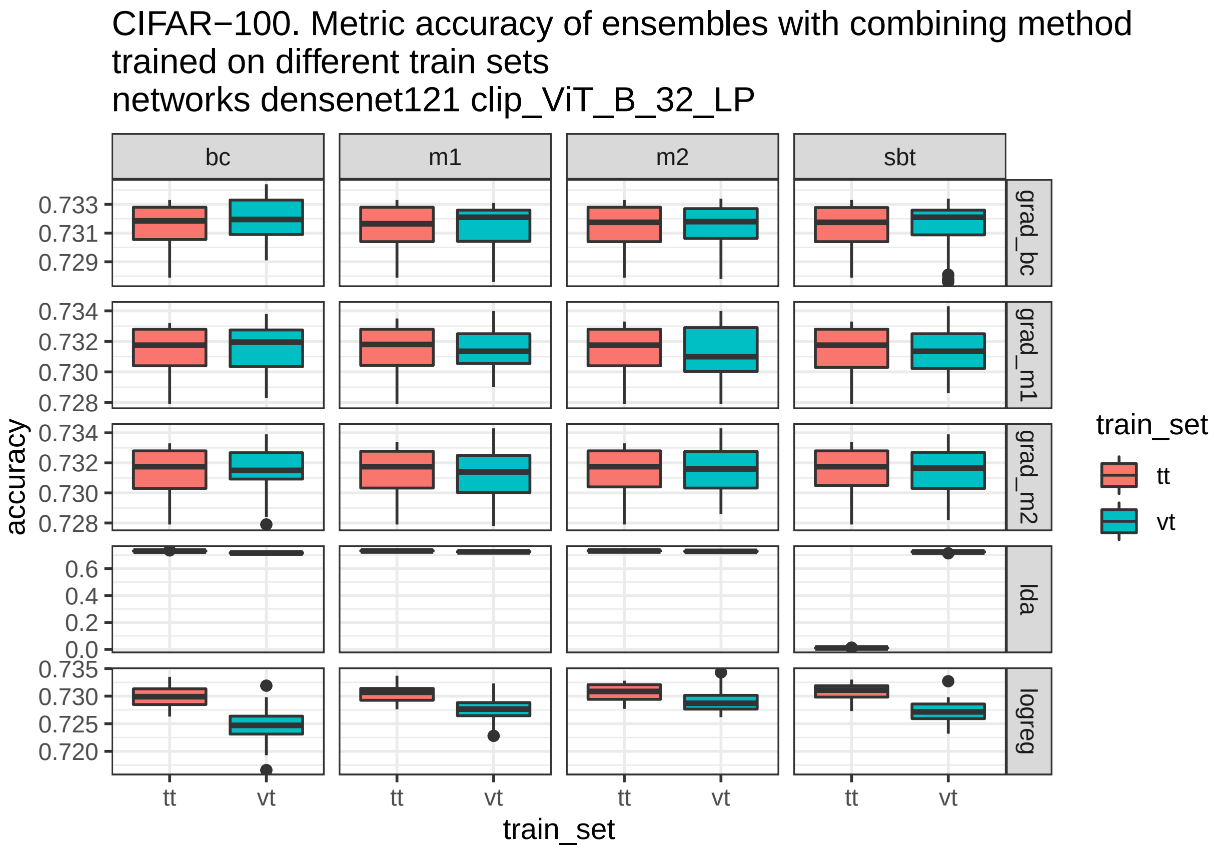


In majority of the combined neural networks subsets, we found that for all coupling methods, the ensemble models trained on subset (of size 500) of the neural networks training set have statistically significantly worse accuracy than ensemble models trained on separate validation set (of the same size). Same is true for metrics NLL and ECE.

CIFAR100

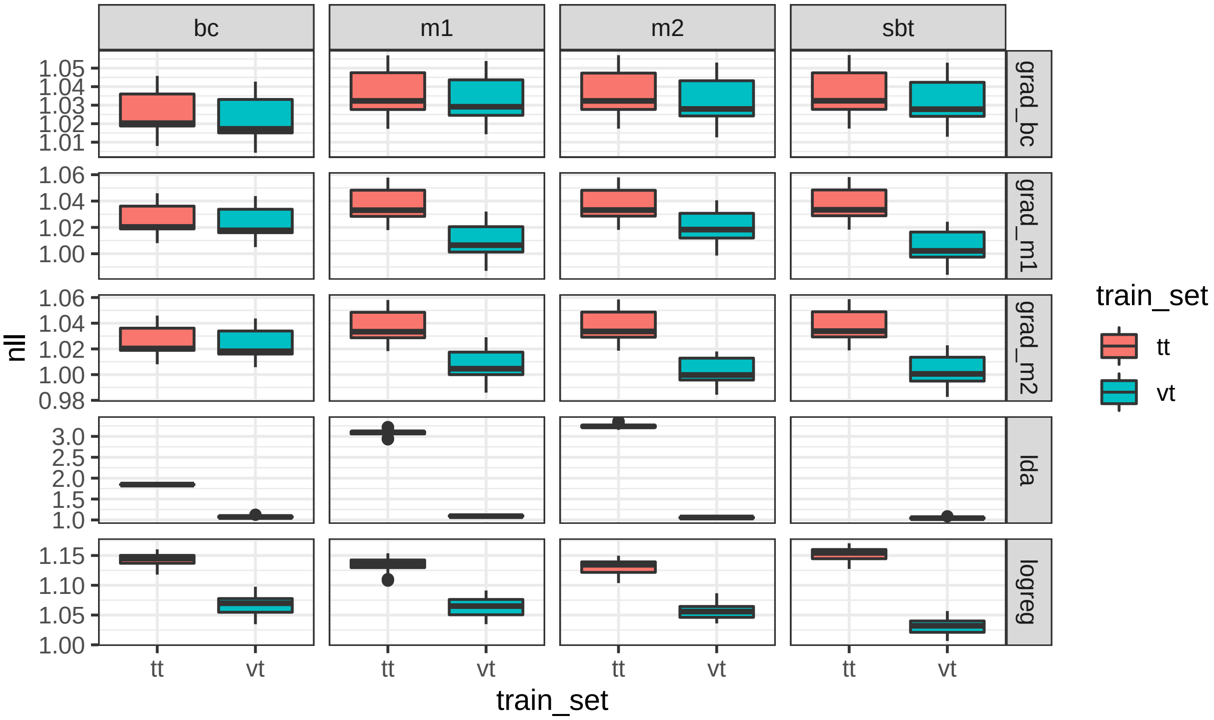
```
net_results_c100 <- read.csv("../data/data_train_val_half_c100/net_metrics.csv")
ens_results_c100 <- read.csv("../data/data_train_val_half_c100/ensemble_metrics.csv")
net_cols <- gsub("-", ".", unique(net_results_c100$network))
```

```
for (comb_id in unique(ens_results_c100$combination_id))
{
    cur_comb_ens_results <- ens_results_c100 %>% filter(combination_id == comb_id)
    comb_nets <- gsub("\\.", "_", net_cols[as.logical(cur_comb_ens_results[1, net_cols])]])
    for (met_i in seq_along(metrics))
    {
        box_plot <- cur_comb_ens_results  %>% ggplot() +
            geom_boxplot(mapping=aes_string(x = "train_set", y = metrics[met_i], fill = "train_set")) +
            facet_grid(rows=vars(combining_method), cols=vars(coupling_method), scales="free") +
            ggtitle(paste0(
                "CIFAR-100. Metric ", metric_names[met_i],
                " of ensembles with combining method\ntrained on different train sets\nnetworks ",
                paste(comb_nets, collapse = " "))) +
            theme_bw()
        print(box_plot)
    }
}
```

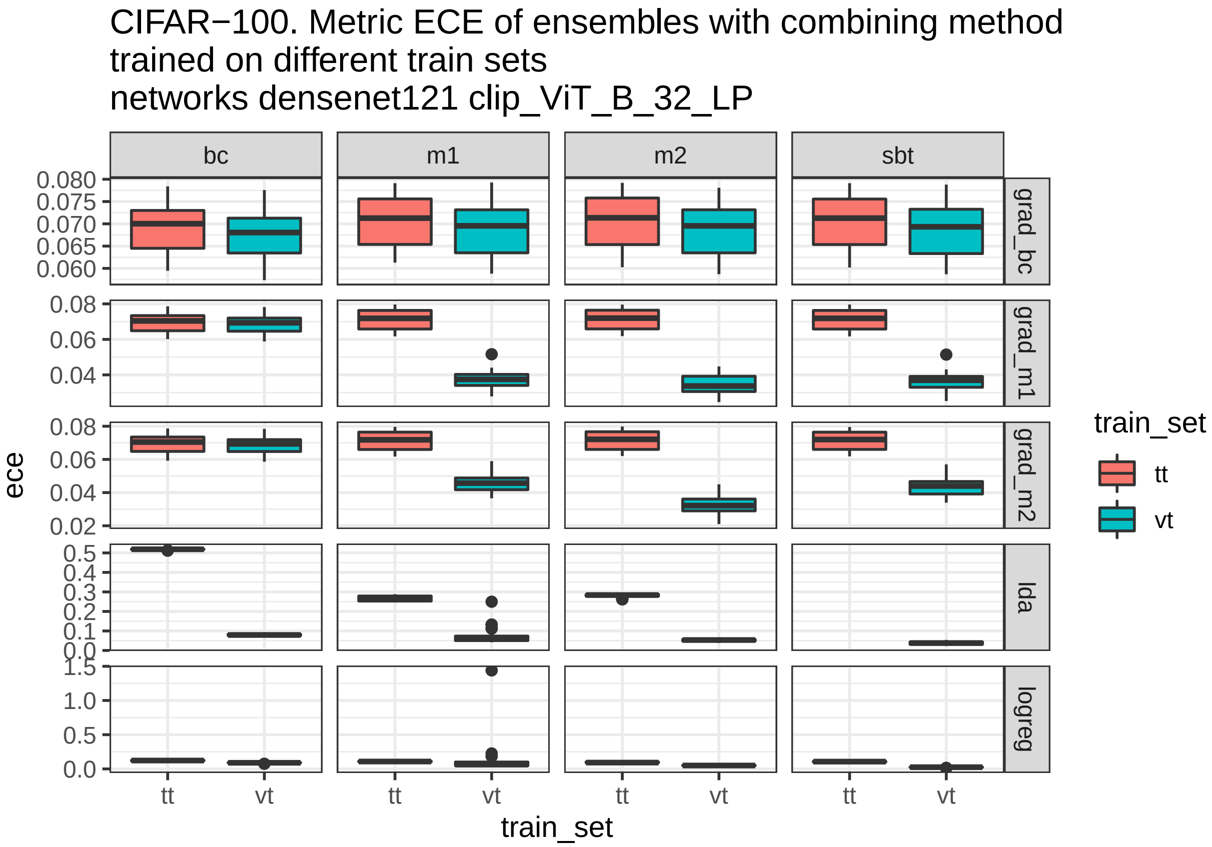


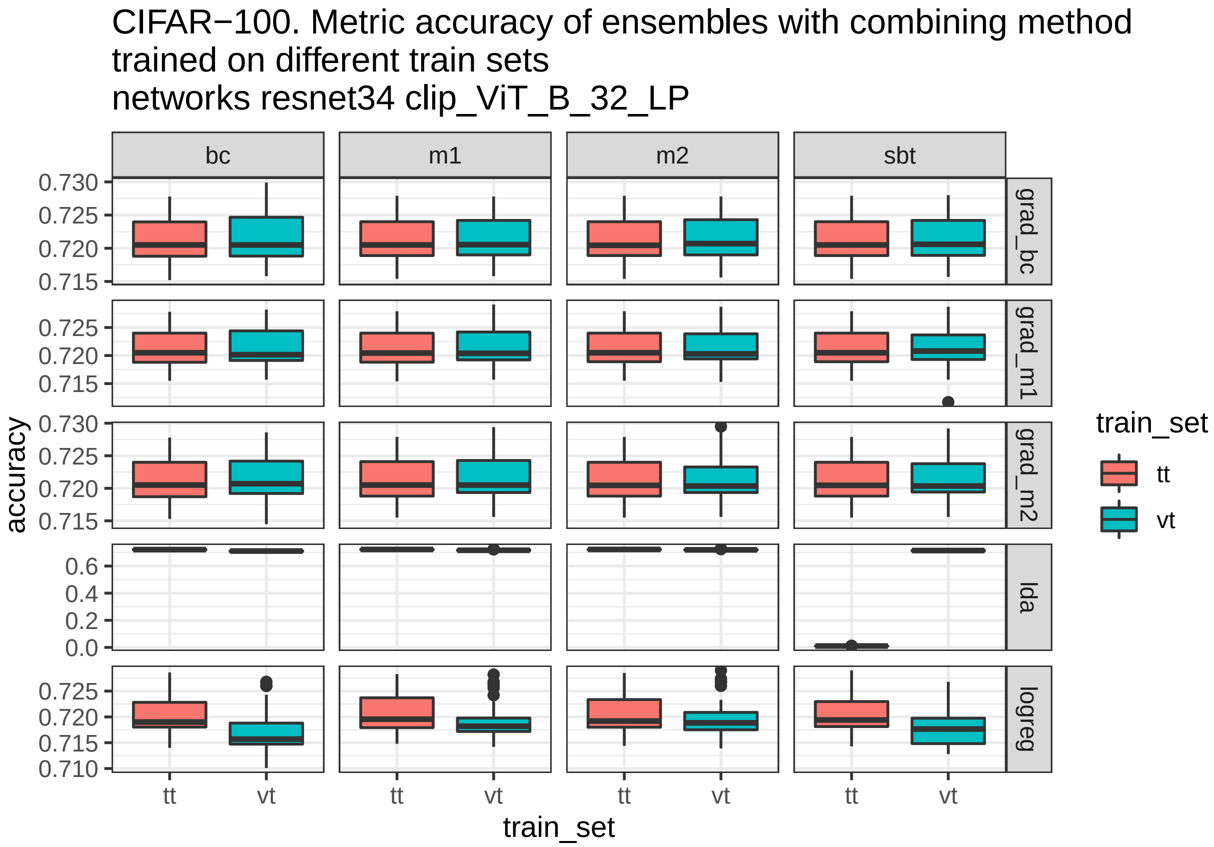
```
## Warning: Removed 60 rows containing non-finite values (stat_boxplot).
```

CIFAR−100. Metric NLL of ensembles with combining method trained on different train sets
networks densenet121 clip_ViT_B_32_LP



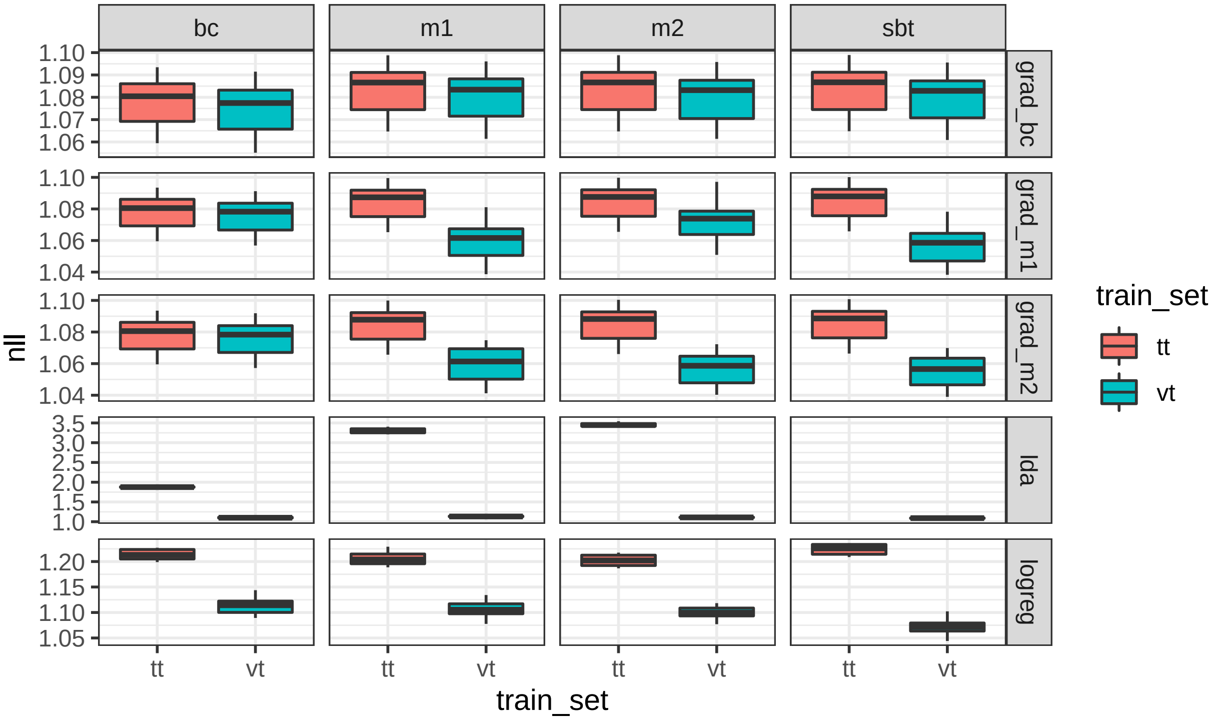
```
## Warning: Removed 60 rows containing non-finite values (stat_boxplot).
```


CIFAR−100. Metric ECE of ensembles with combining method trained on different train sets
networks densenet121 clip_ViT_B_32_LP
bc
m1
m2
sbt
grad_bc
grad_m1
grad_m2
lda
logreg
ece
0.080
0.075
0.070
0.065
0.060
0.08
0.06
0.04
0.08
0.06
0.04
0.02
0.5
0.4
0.3
0.2
0.1
0.0
1.5
1.0
0.5
0.0
tt
vt
train_set
train_set
tt
vt



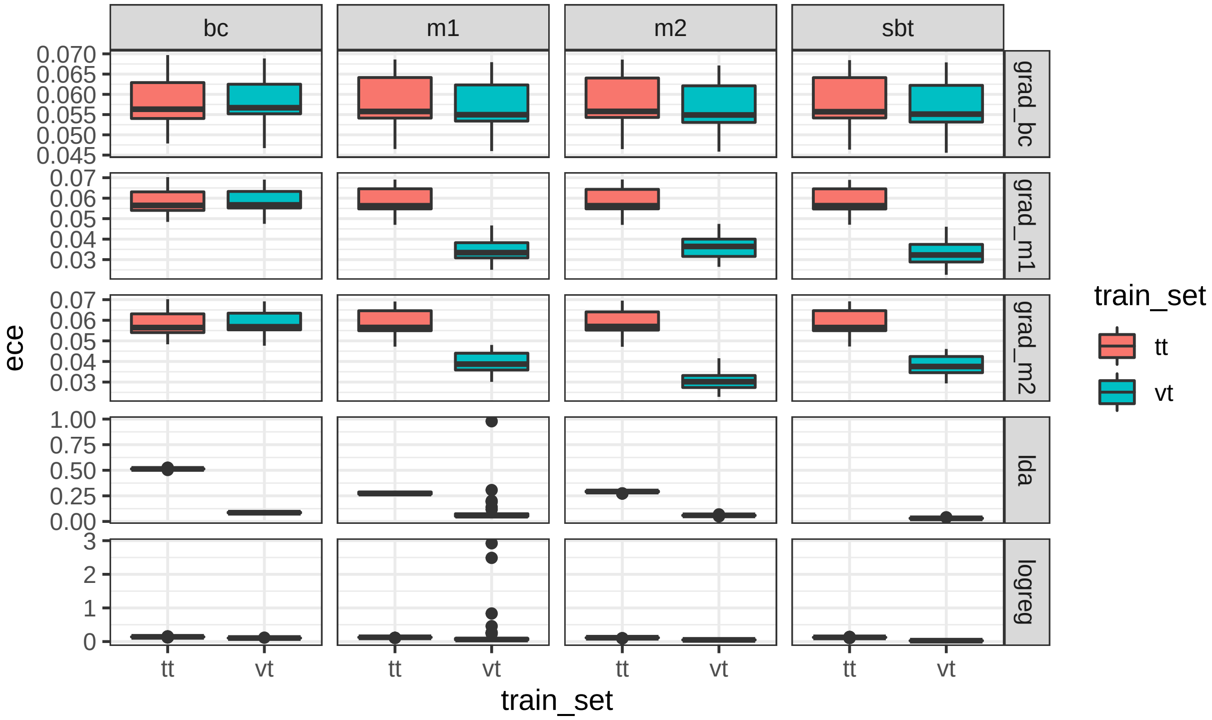
```
## Warning: Removed 52 rows containing non-finite values (stat_boxplot).
```

CIFAR−100. Metric NLL of ensembles with combining method
trained on different train sets
networks resnet34 clip_ViT_B_32_LP

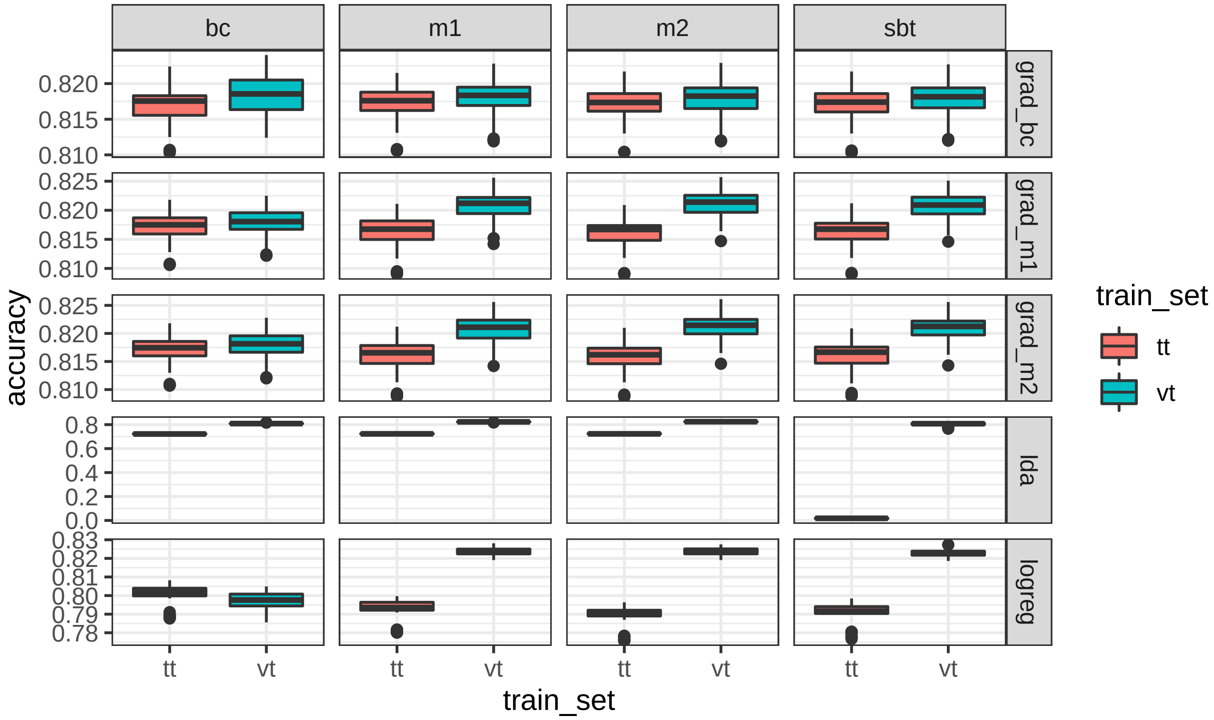


```
## Warning: Removed 52 rows containing non-finite values (stat_boxplot).
```

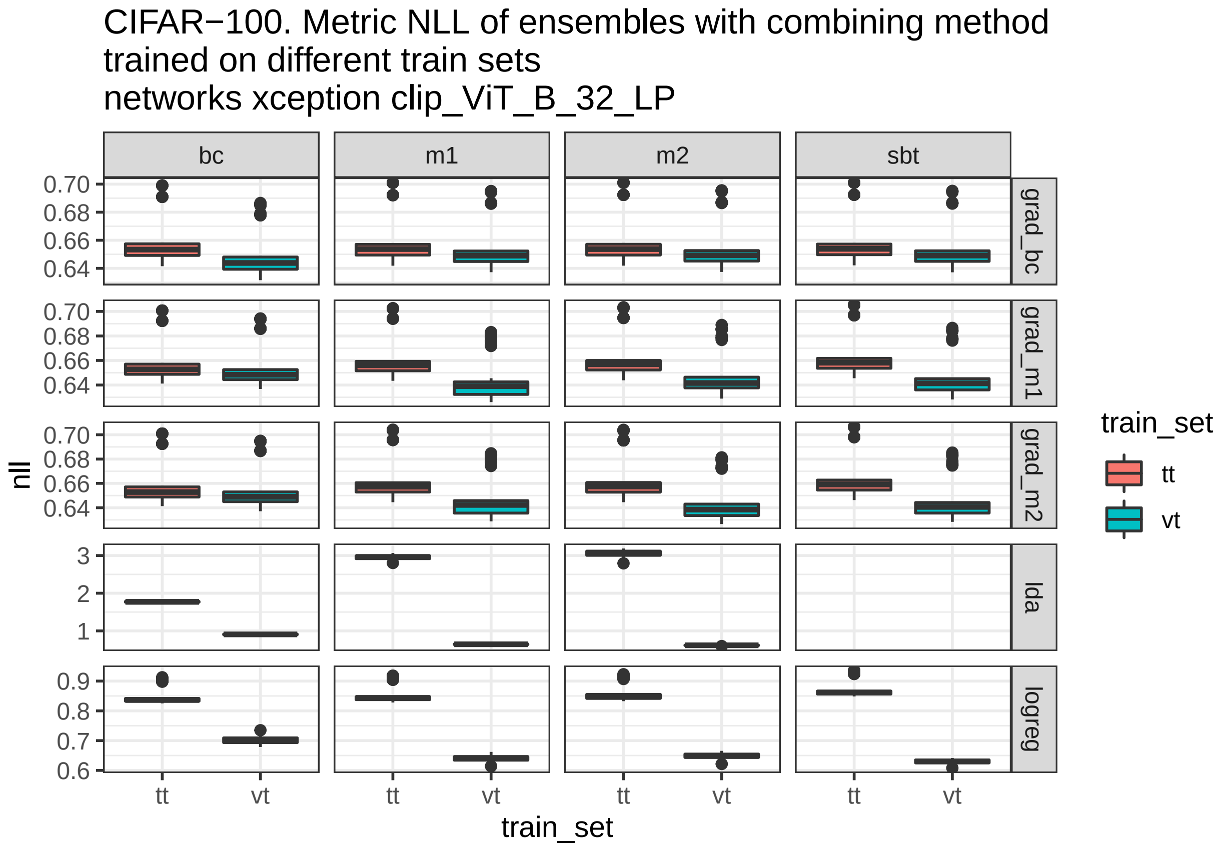
CIFAR−100. Metric ECE of ensembles with combining method trained on different train sets
networks resnet34 clip_ViT_B_32_LP



CIFAR−100. Metric accuracy of ensembles with combining method
trained on different train sets
networks xception clip_ViT_B_32_LP

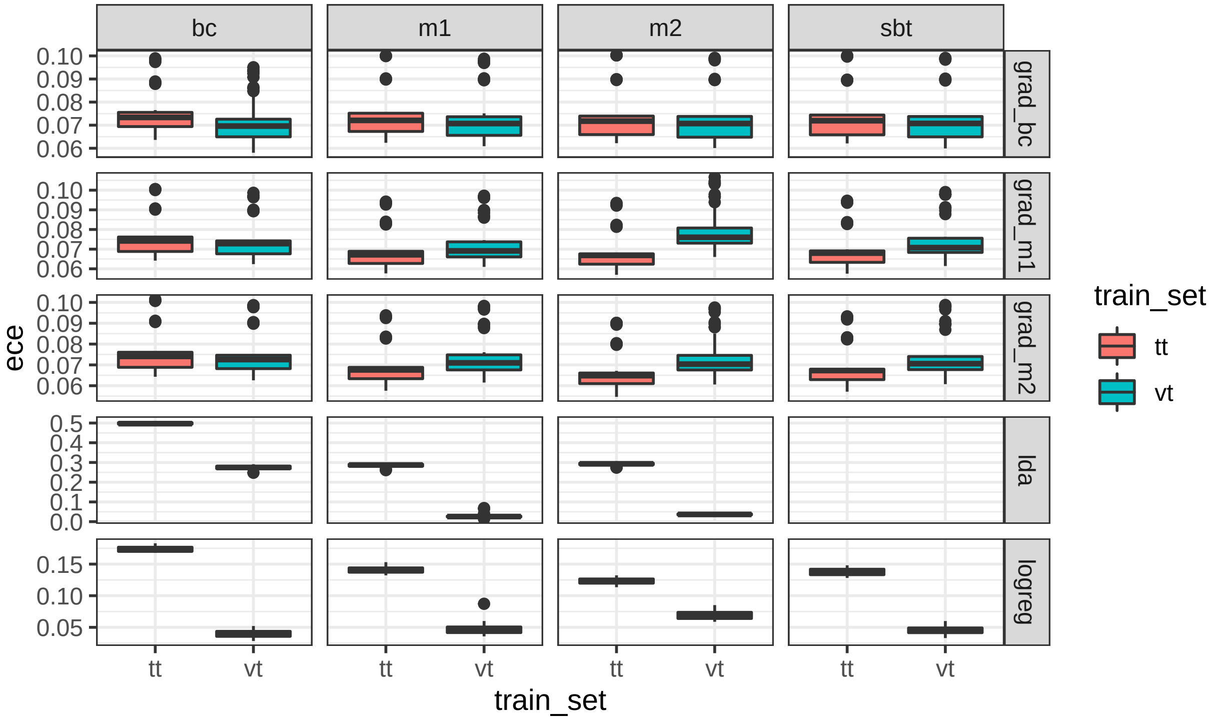


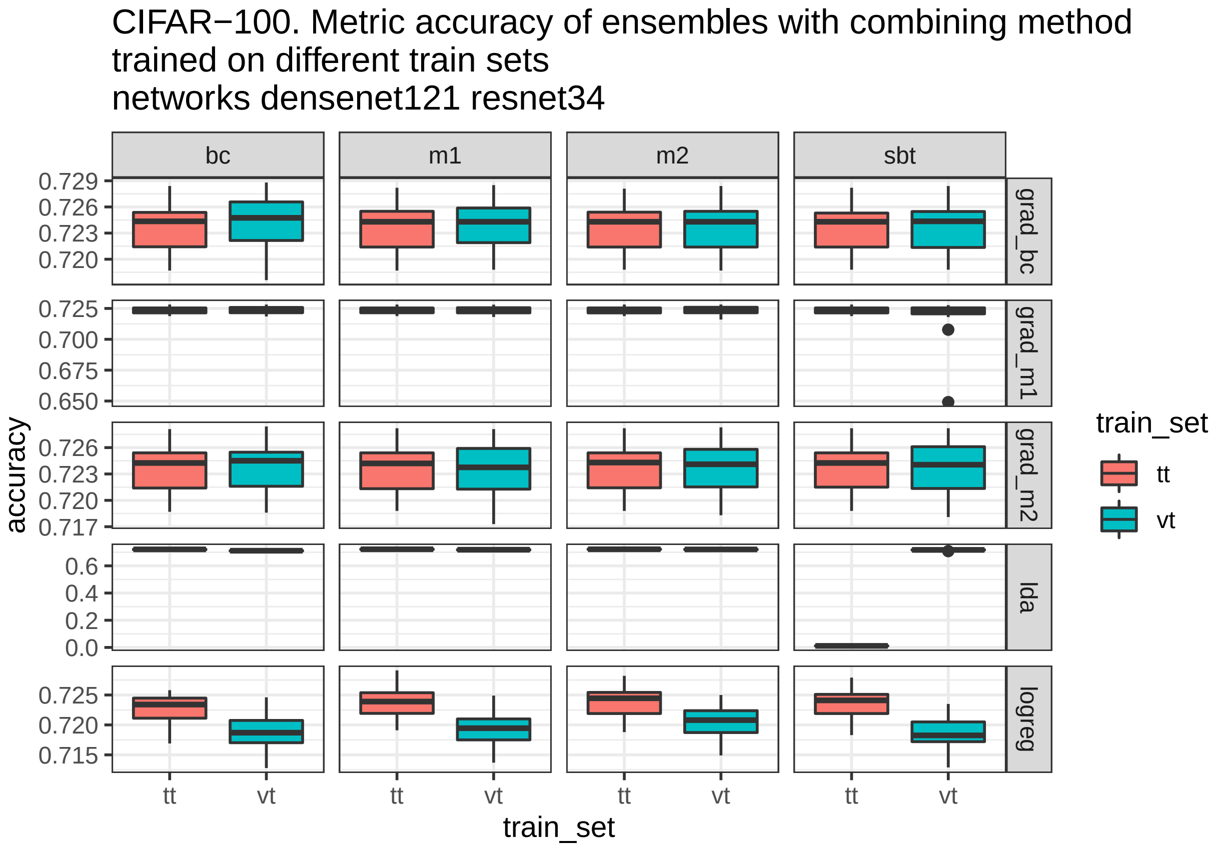
```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```



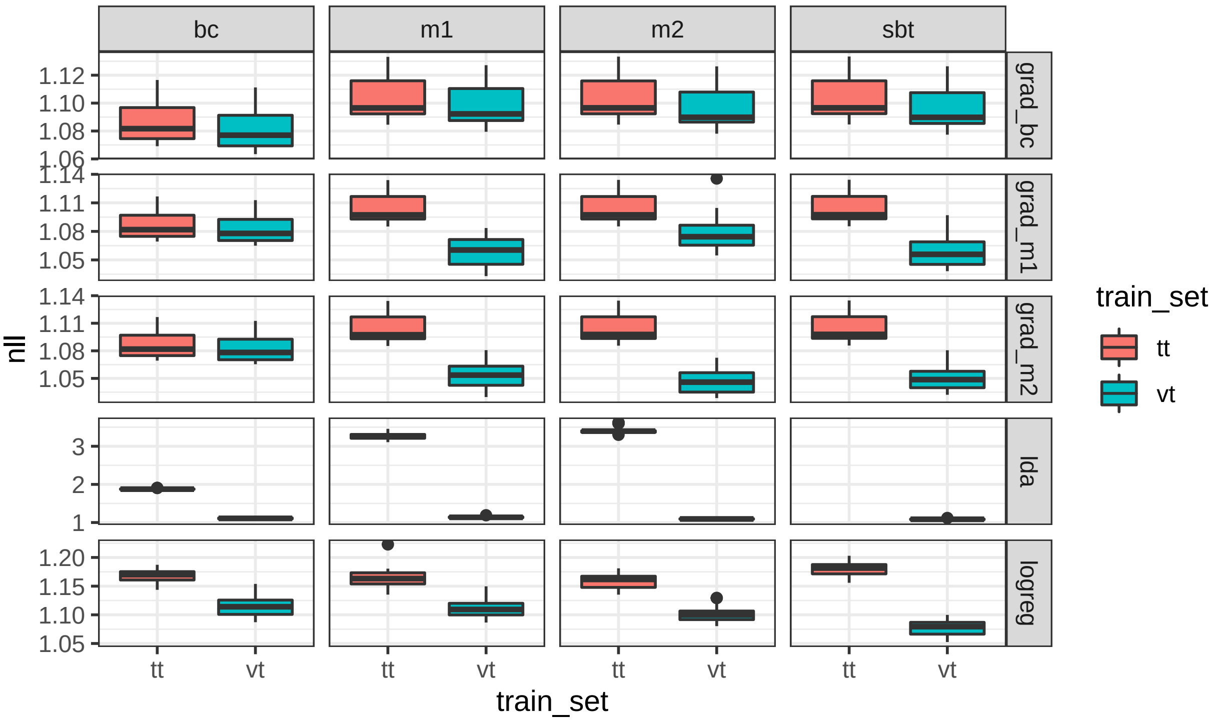
```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```

CIFAR−100. Metric ECE of ensembles with combining method
trained on different train sets
networks xception clip_ViT_B_32_LP





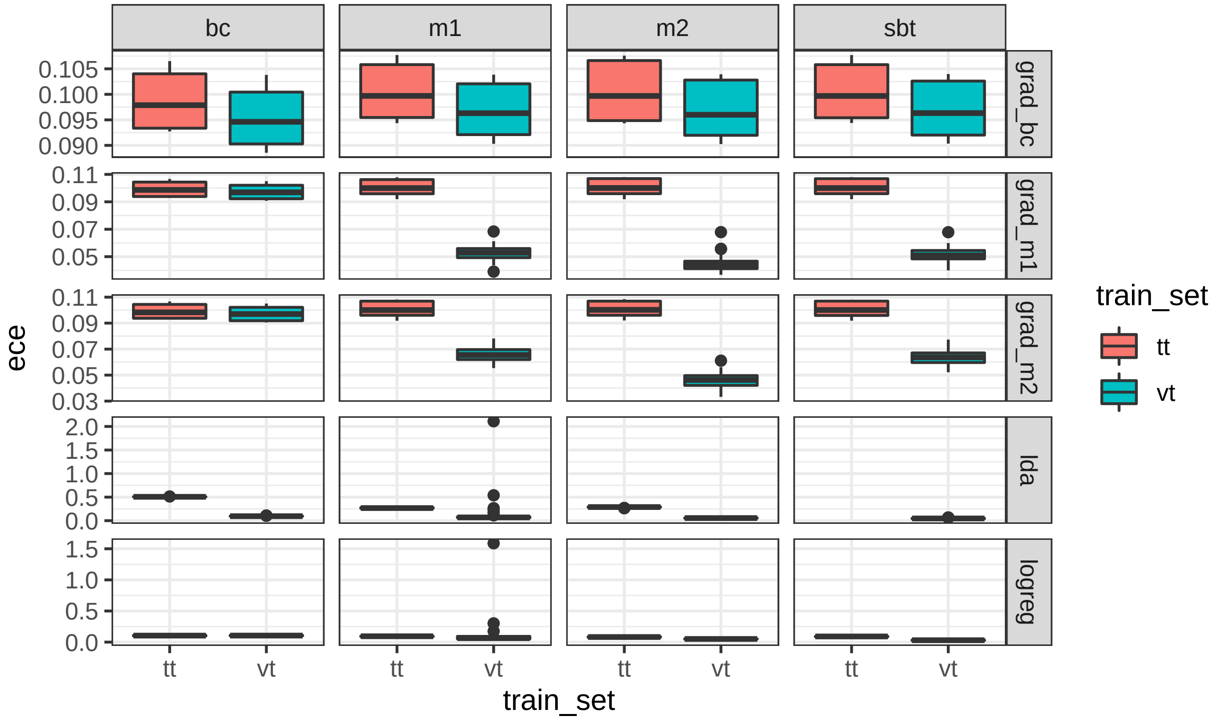
```
## Warning: Removed 64 rows containing non-finite values (stat_boxplot).
```

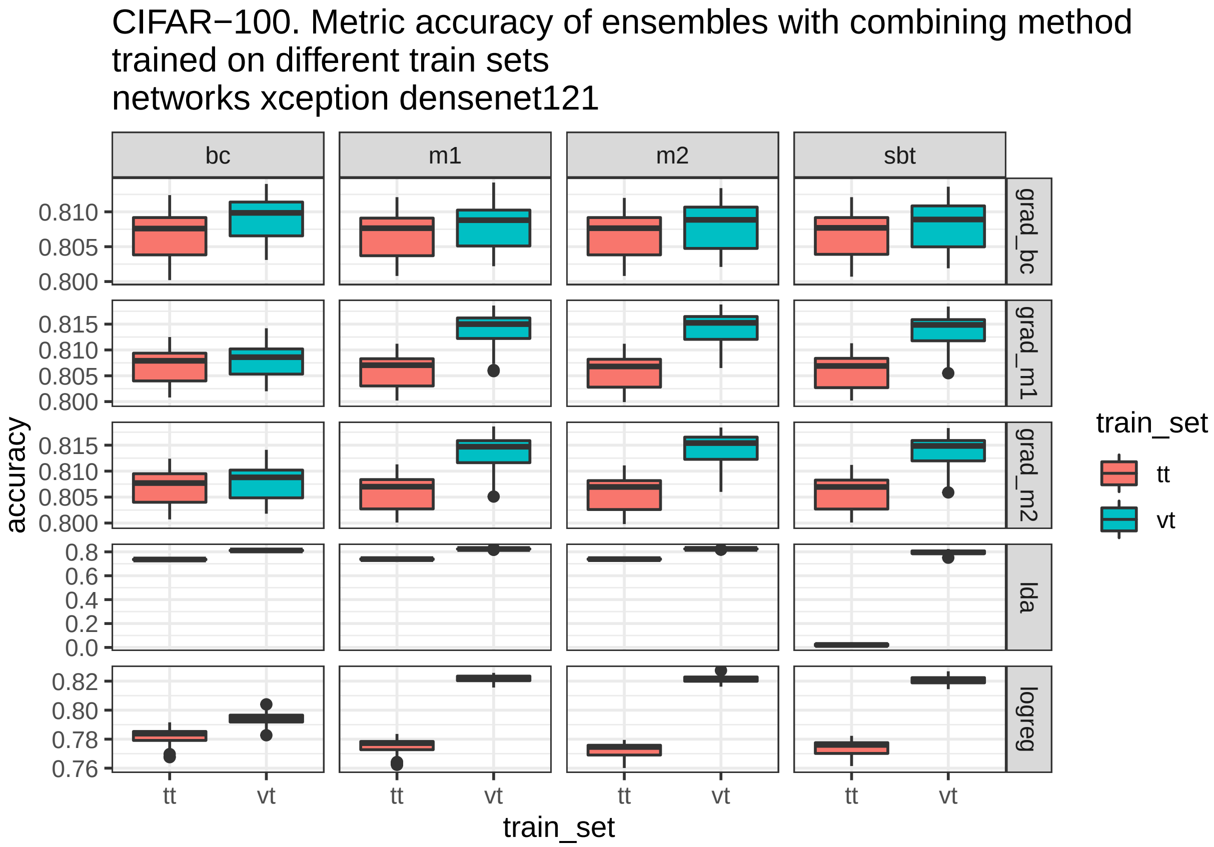


CIFAR−100. Metric NLL of ensembles with combining method trained on different train sets
networks densenet121 resnet34

```
## Warning: Removed 64 rows containing non-finite values (stat_boxplot).
```

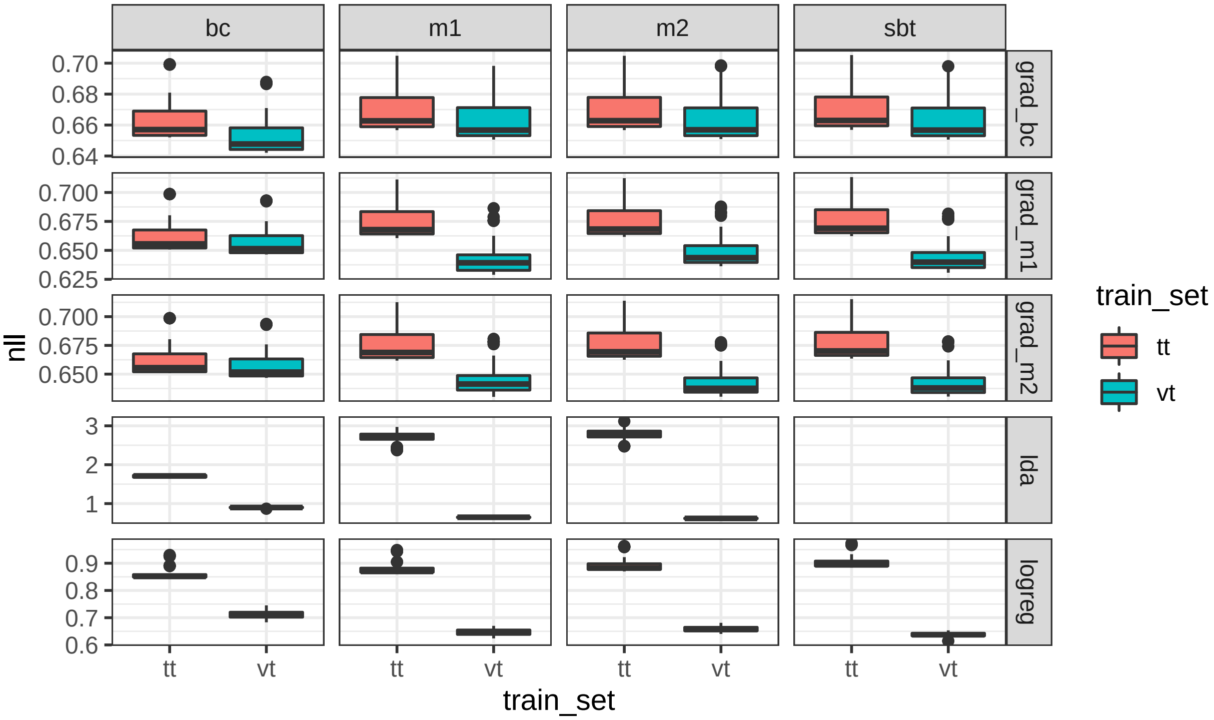
CIFAR−100. Metric ECE of ensembles with combining method
trained on different train sets
networks densenet121 resnet34





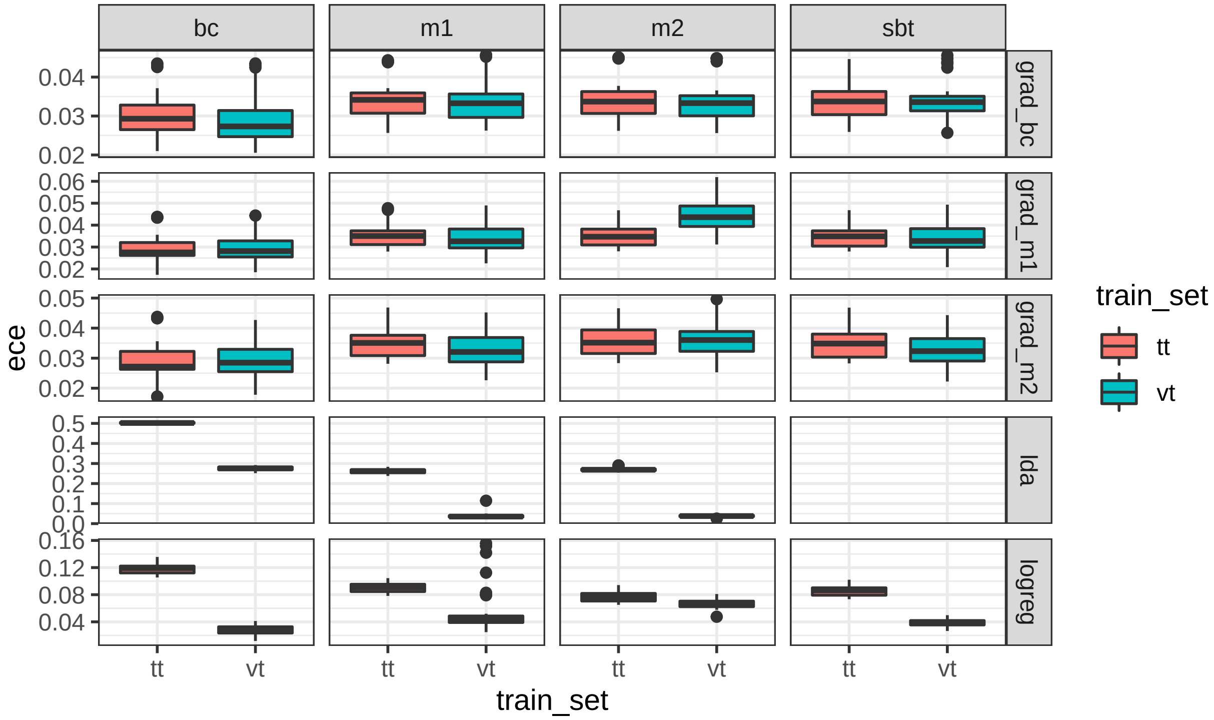
```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```

CIFAR−100. Metric NLL of ensembles with combining method
trained on different train sets
networks xception densenet121

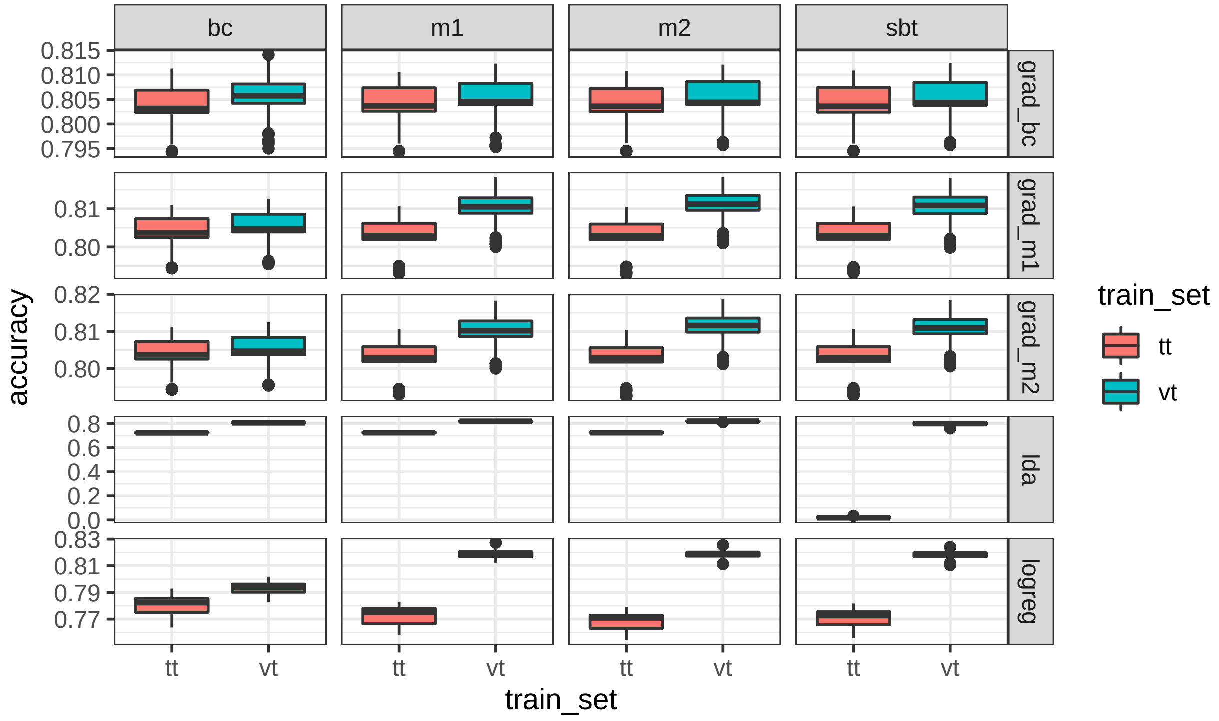


```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```

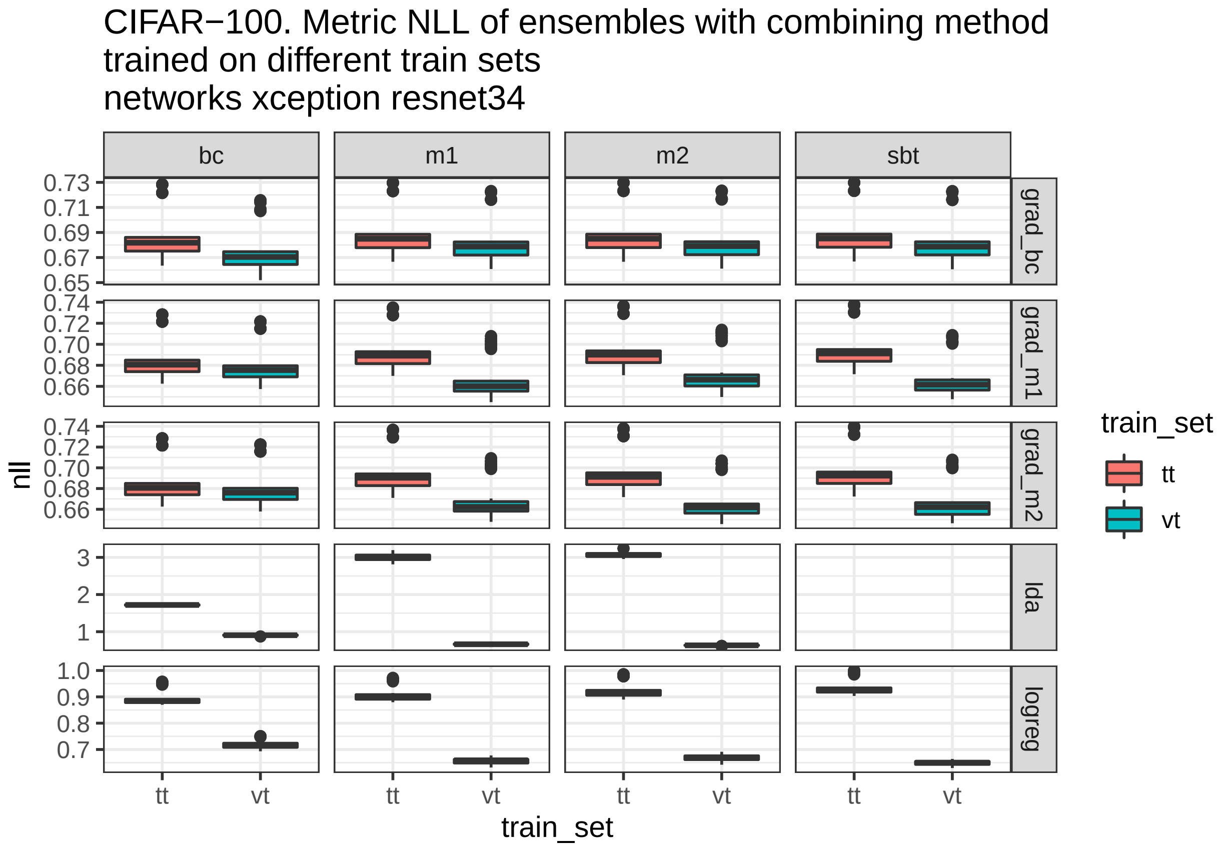
CIFAR−100. Metric ECE of ensembles with combining method
trained on different train sets
networks xception densenet121



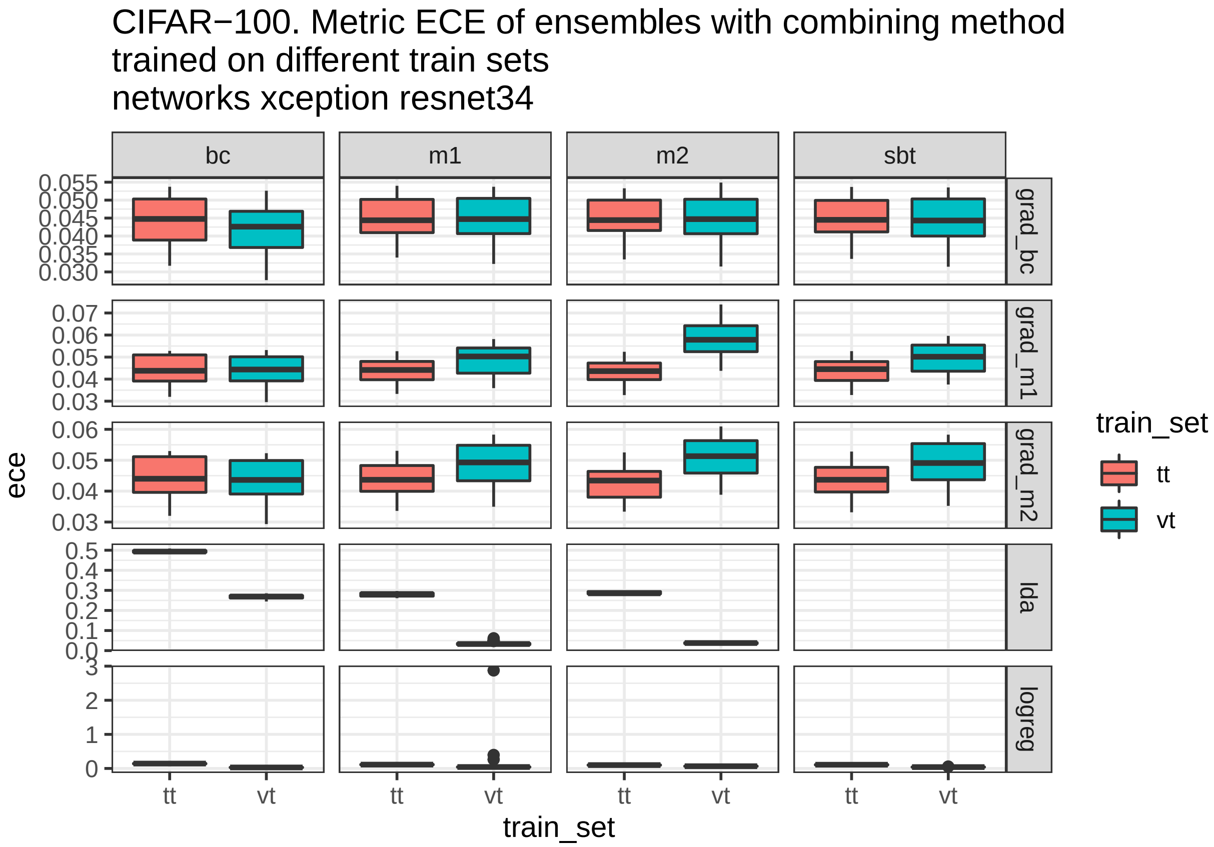
CIFAR−100. Metric accuracy of ensembles with combining method
trained on different train sets
networks xception resnet34

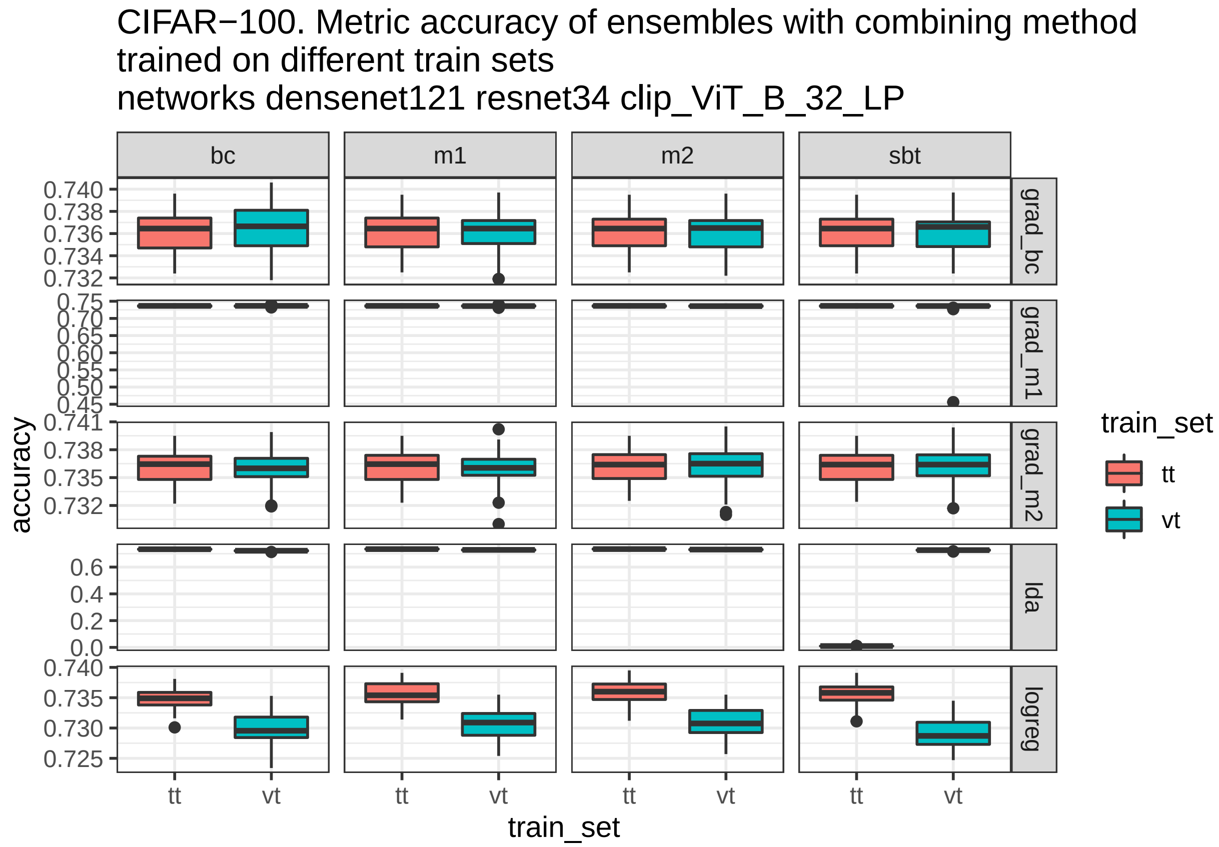


```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```

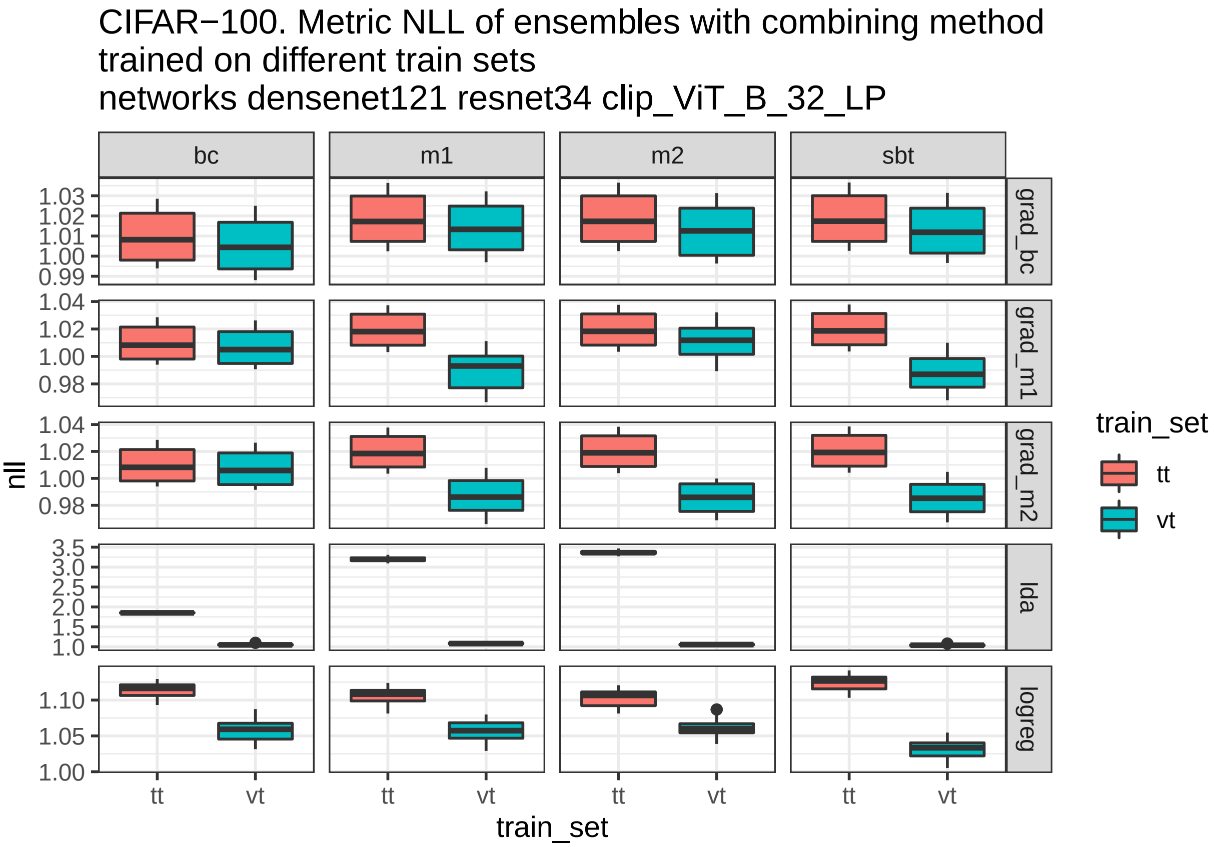


```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```

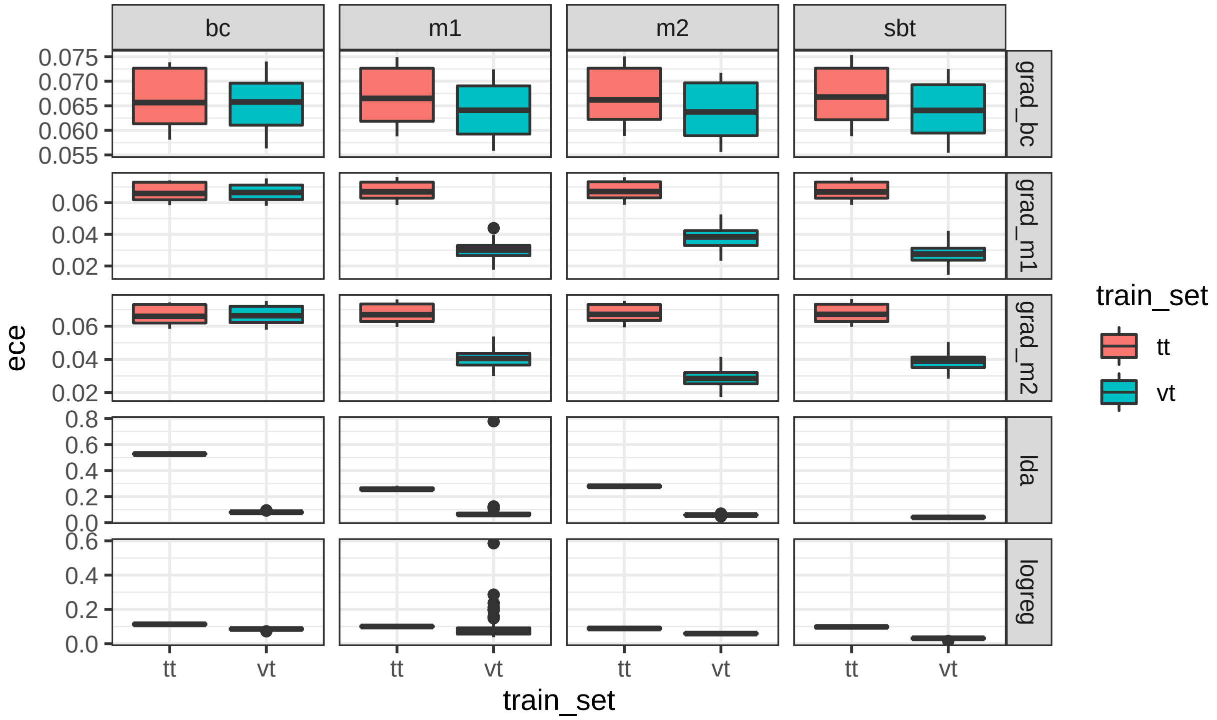

CIFAR−100. Metric ECE of ensembles with combining method
trained on different train sets
networks xception resnet34
bc
m1
m2
sbt
grad_bc
grad_m1
grad_m2
lda
logreg
0.055
0.050
0.045
0.040
0.035
0.030
0.07
0.06
0.05
0.04
0.03
0.06
0.05
0.04
0.03
0.5
0.4
0.3
0.2
0.1
0.0
3
2
1
0
ece
tt
vt
train_set
train_set
tt
vt



```
## Warning: Removed 64 rows containing non-finite values (stat_boxplot).
```

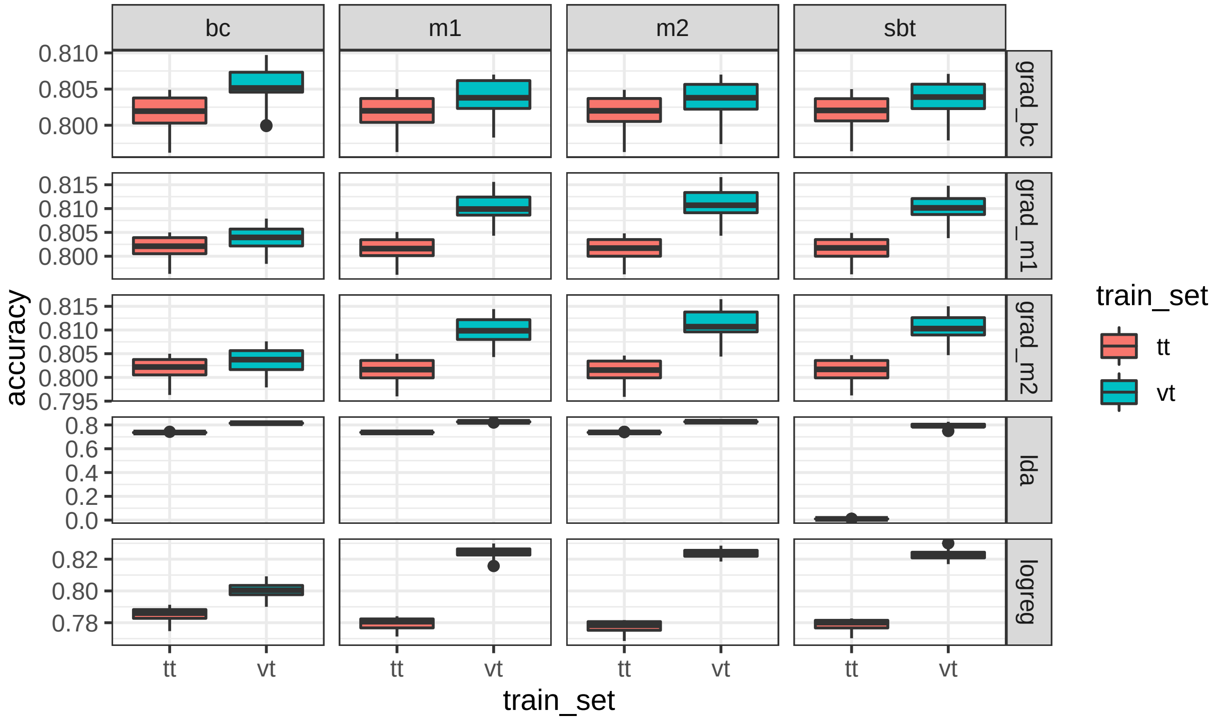


```
## Warning: Removed 64 rows containing non-finite values (stat_boxplot).
```

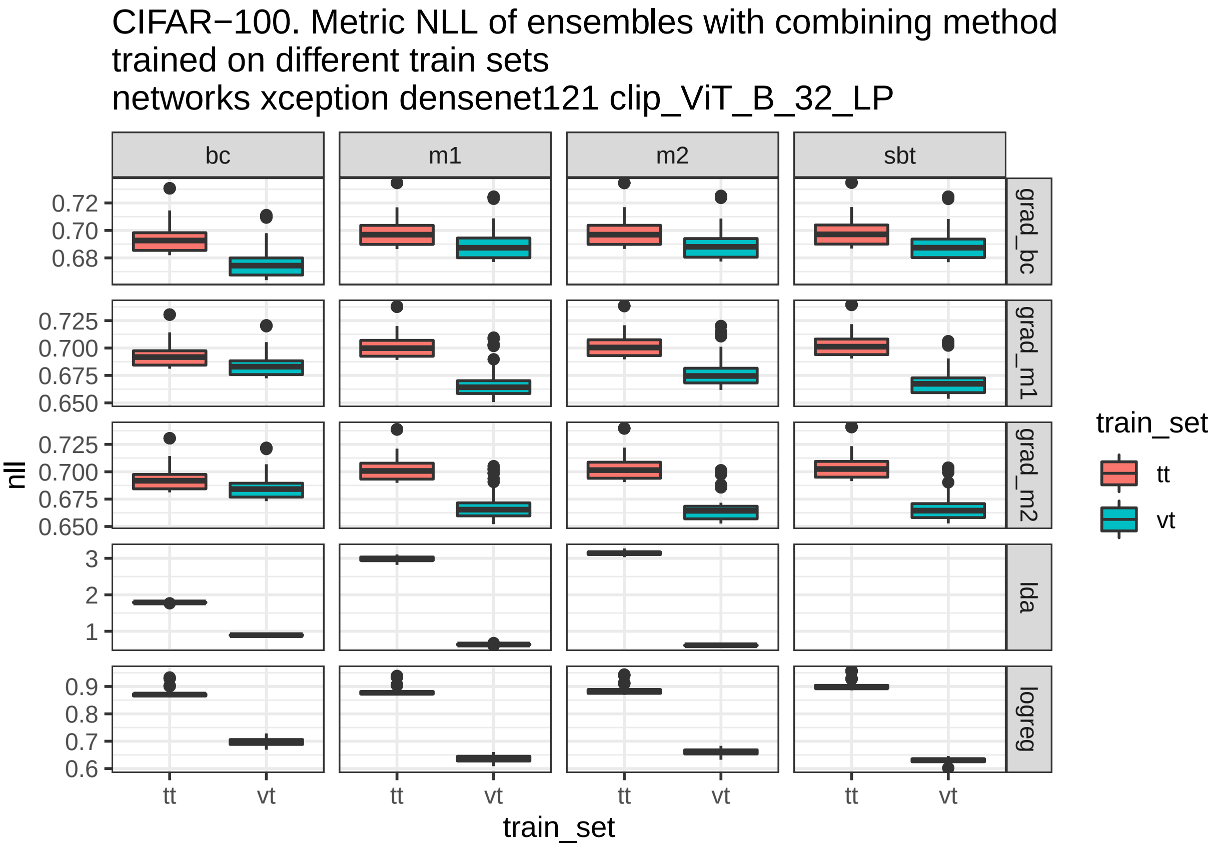


CIFAR−100. Metric ECE of ensembles with combining method trained on different train sets
networks densenet121 resnet34 clip_ViT_B_32_LP

CIFAR−100. Metric accuracy of ensembles with combining method trained on different train sets
networks xception densenet121 clip_ViT_B_32_LP

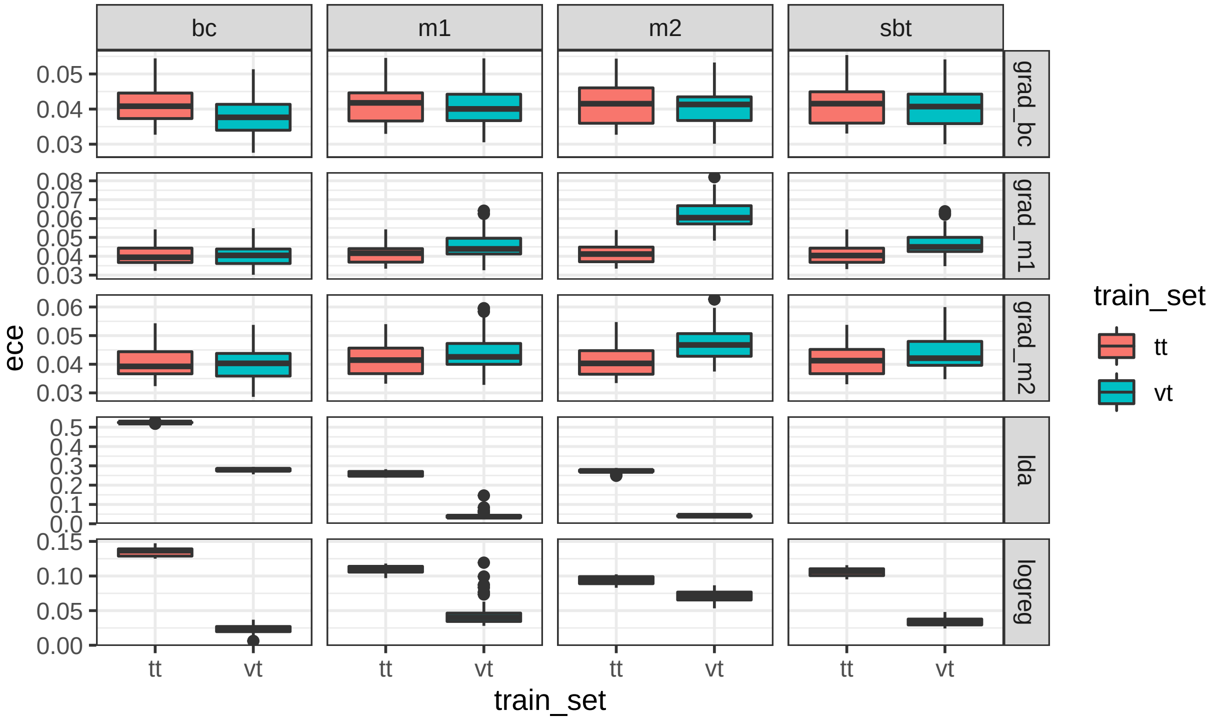


```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```

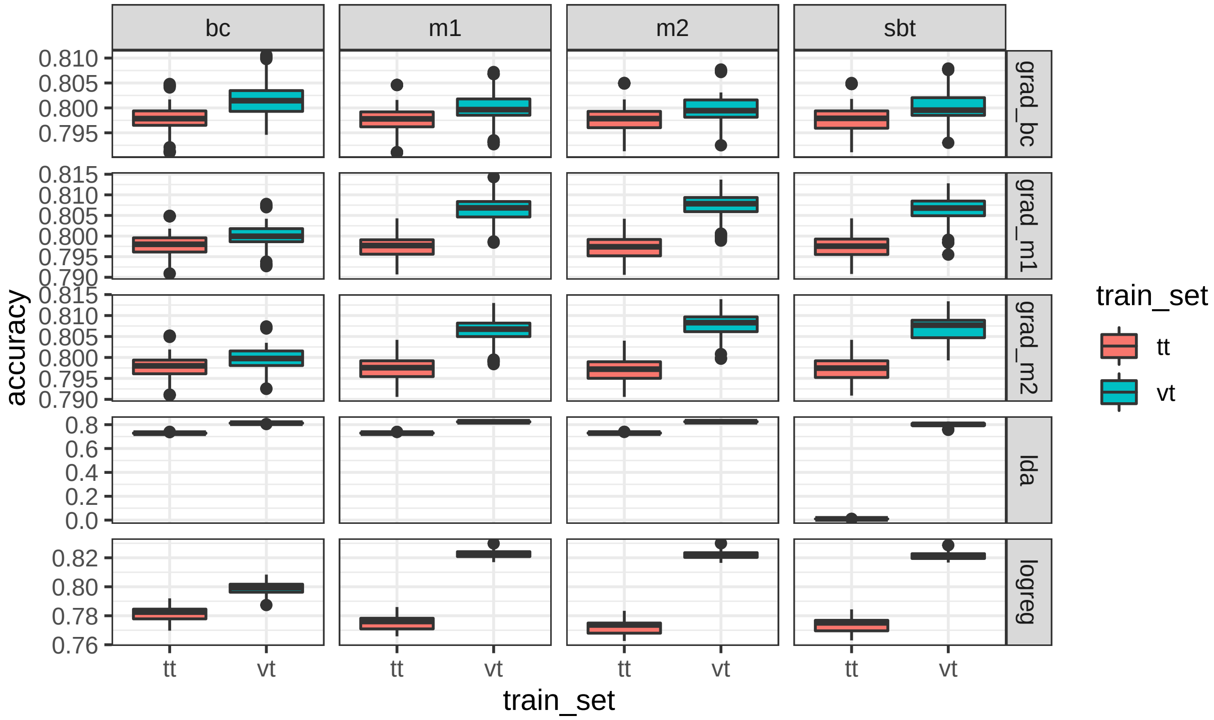


```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```

CIFAR−100. Metric ECE of ensembles with combining method trained on different train sets
networks xception densenet121 clip_ViT_B_32_LP

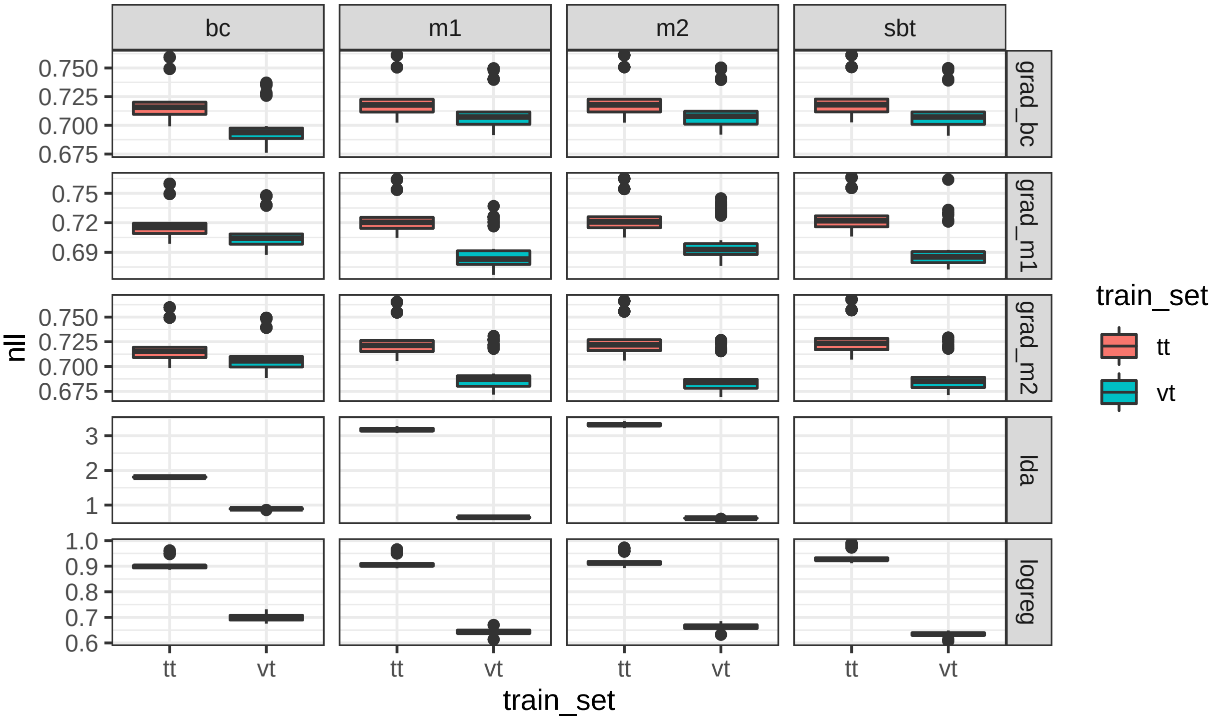


CIFAR−100. Metric accuracy of ensembles with combining method trained on different train sets
networks xception resnet34 clip_ViT_B_32_LP

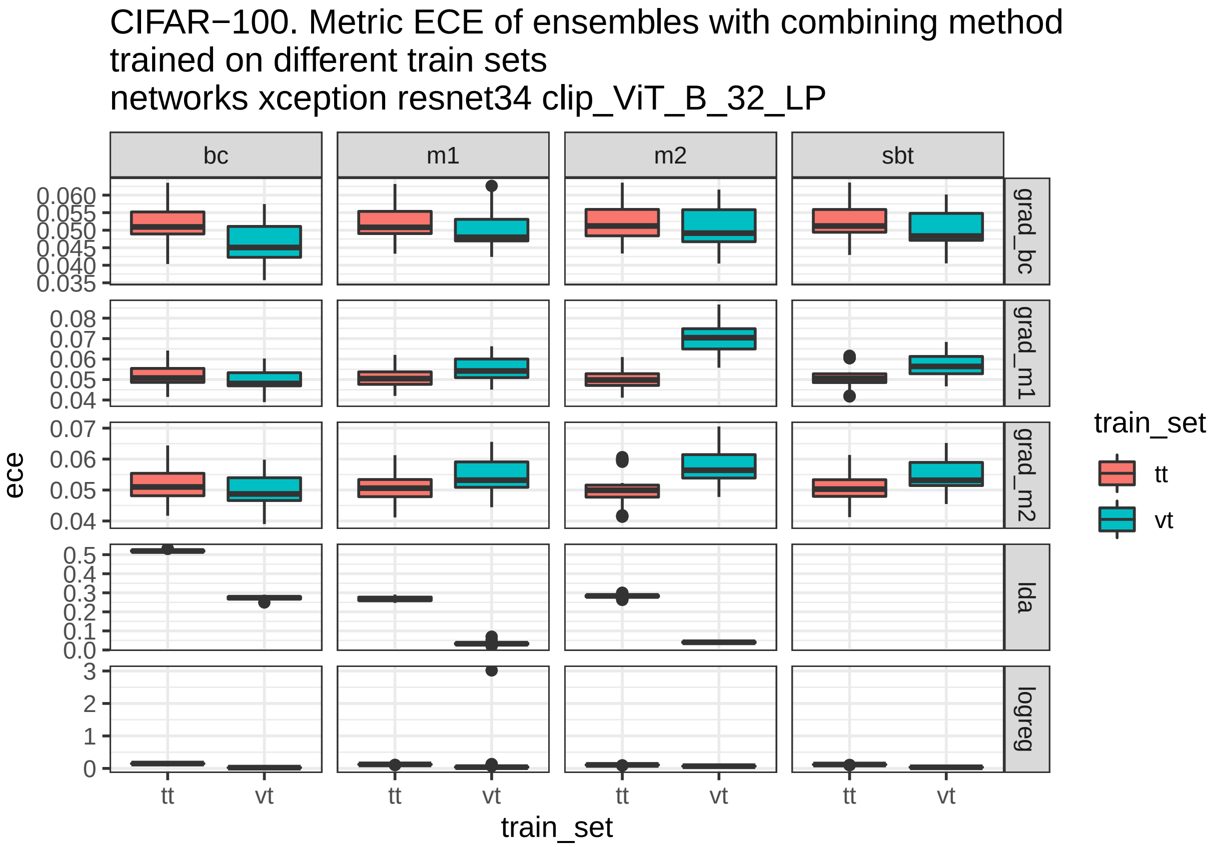


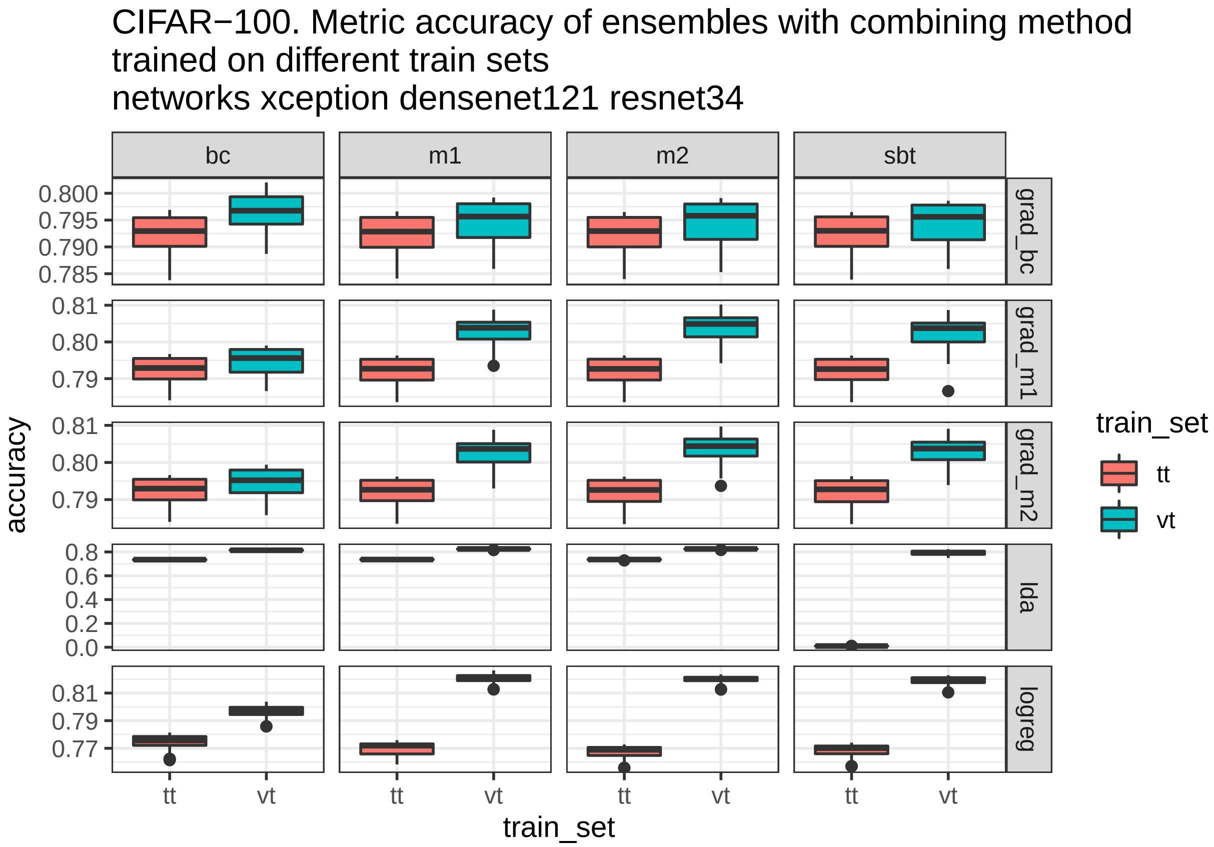
```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```

CIFAR−100. Metric NLL of ensembles with combining method trained on different train sets
networks xception resnet34 clip_ViT_B_32_LP



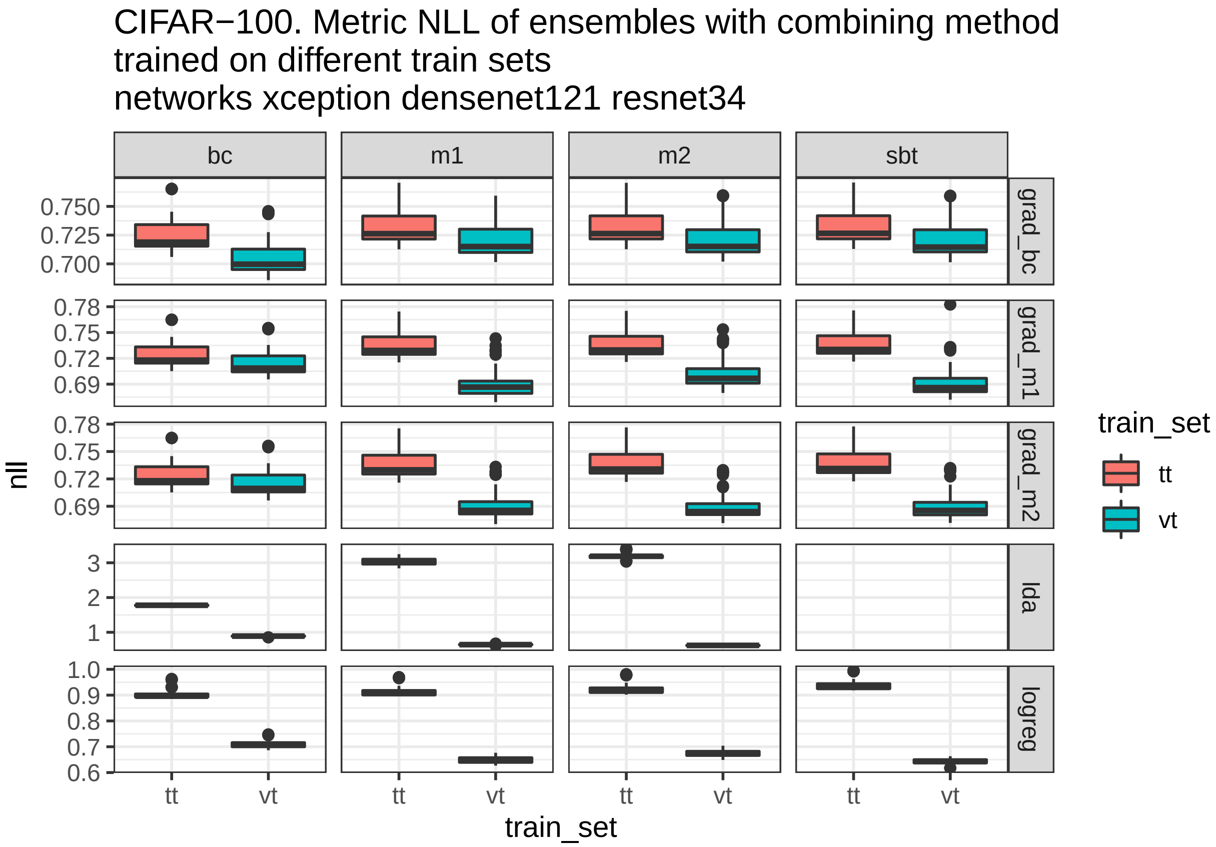
```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```


CIFAR−100. Metric ECE of ensembles with combining method
trained on different train sets
networks xception resnet34 clip_ViT_B_32_LP
bc
m1
m2
sbt
grad_bc
grad_m1
grad_m2
lda
logreg
0.060
0.055
0.050
0.045
0.040
0.035
0.08
0.07
0.06
0.05
0.04
0.07
0.06
0.05
0.04
0.5
0.4
0.3
0.2
0.1
0.0
3
2
1
0
ece
tt
vt
train_set
train_set
tt
vt



CIFAR−100. Metric accuracy of ensembles with combining method
trained on different train sets
networks xception densenet121 resnet34

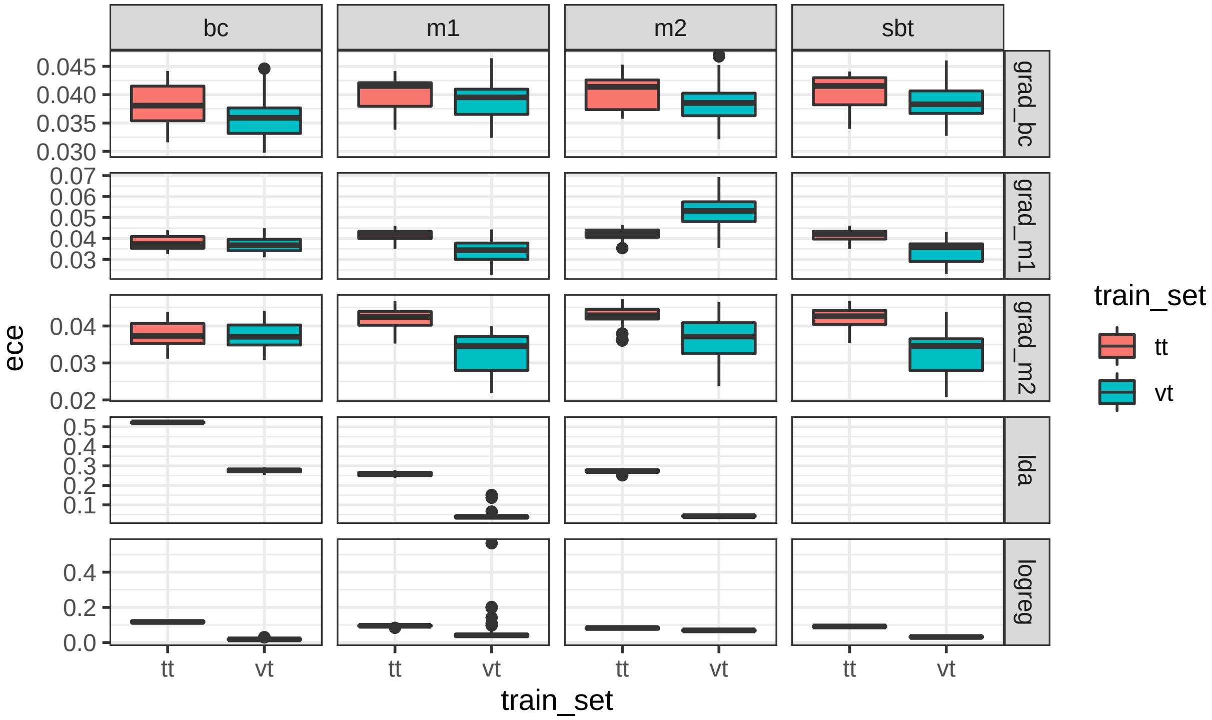
```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```



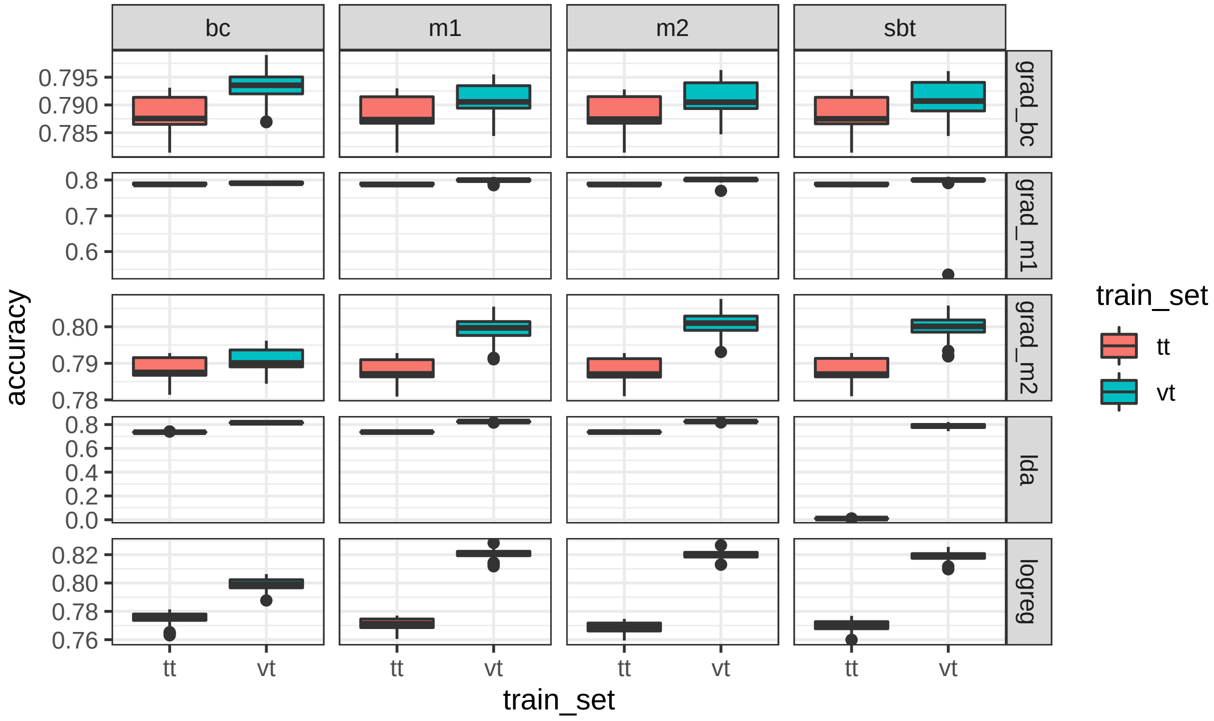
CIFAR−100. Metric NLL of ensembles with combining method trained on different train sets
networks xception densenet121 resnet34

```
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```

CIFAR−100. Metric ECE of ensembles with combining method trained on different train sets
networks xception densenet121 resnet34

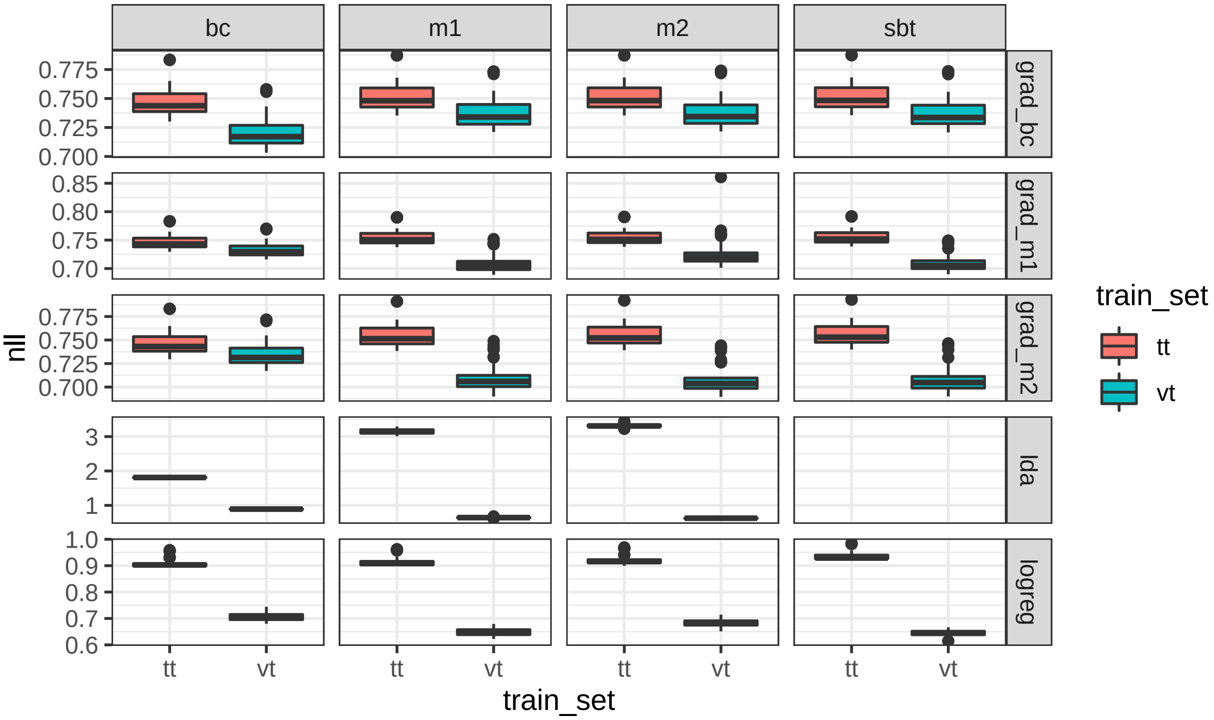


CIFAR−100. Metric accuracy of ensembles with combining method trained on different train sets
networks xception densenet121 resnet34 clip_ViT_B_32_LP

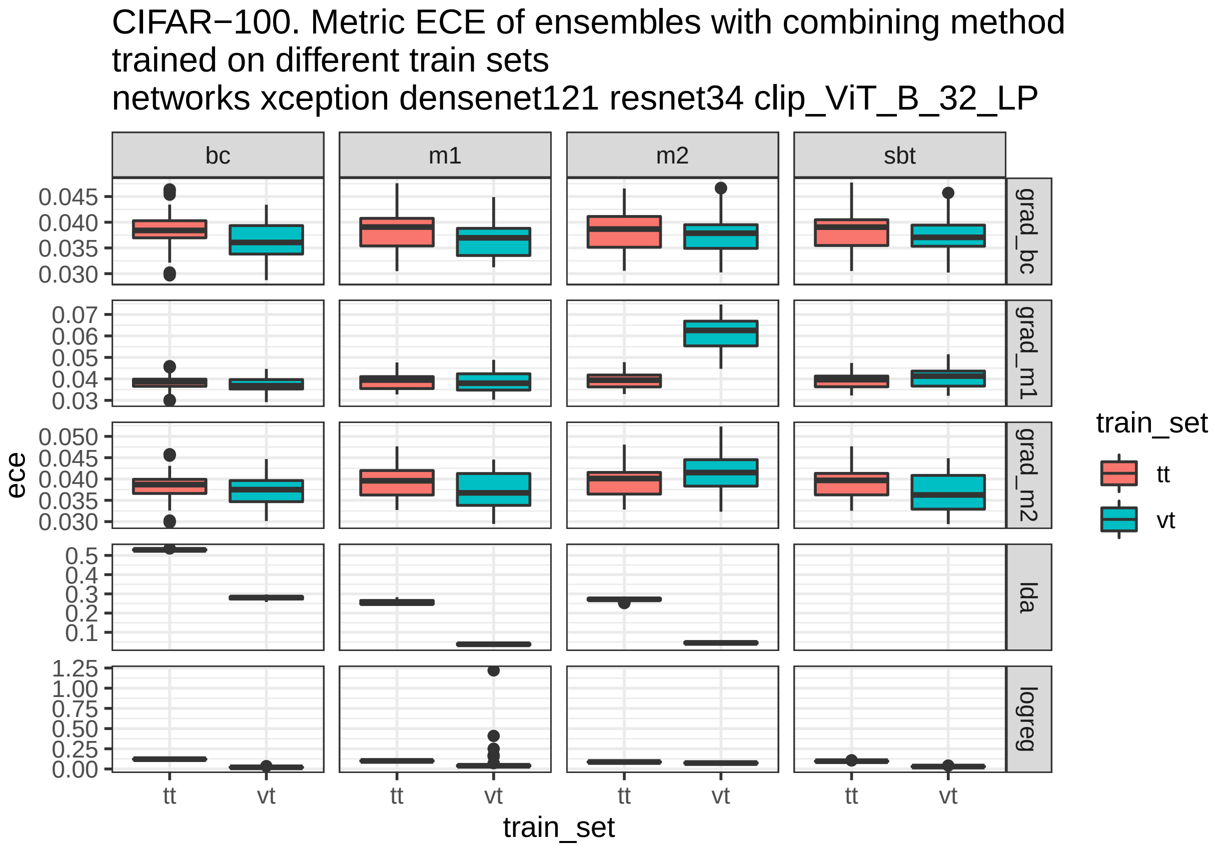


```
## Warning: Removed 101 rows containing non-finite values (stat_boxplot).
```

CIFAR−100. Metric NLL of ensembles with combining method trained on different train sets
networks xception densenet121 resnet34 clip_ViT_B_32_LP

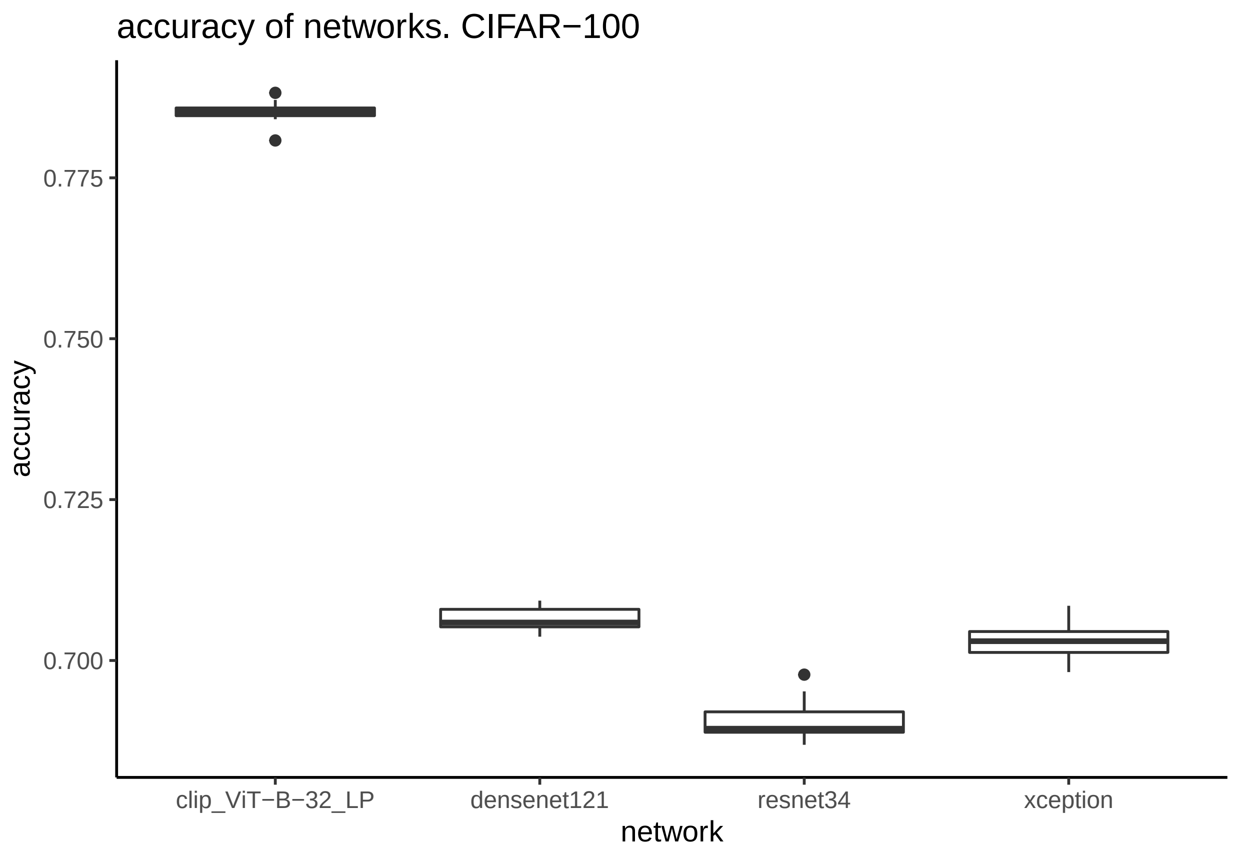


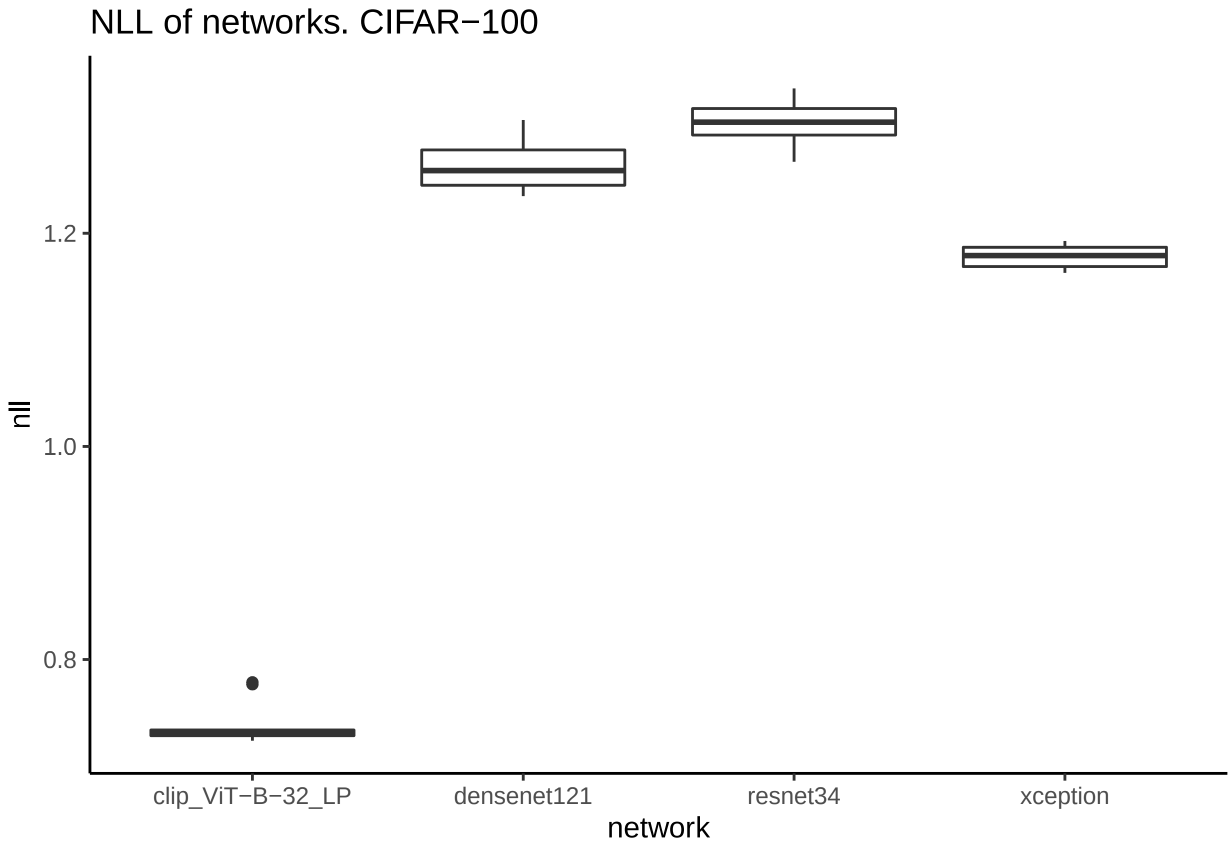
```
## Warning: Removed 101 rows containing non-finite values (stat_boxplot).
```

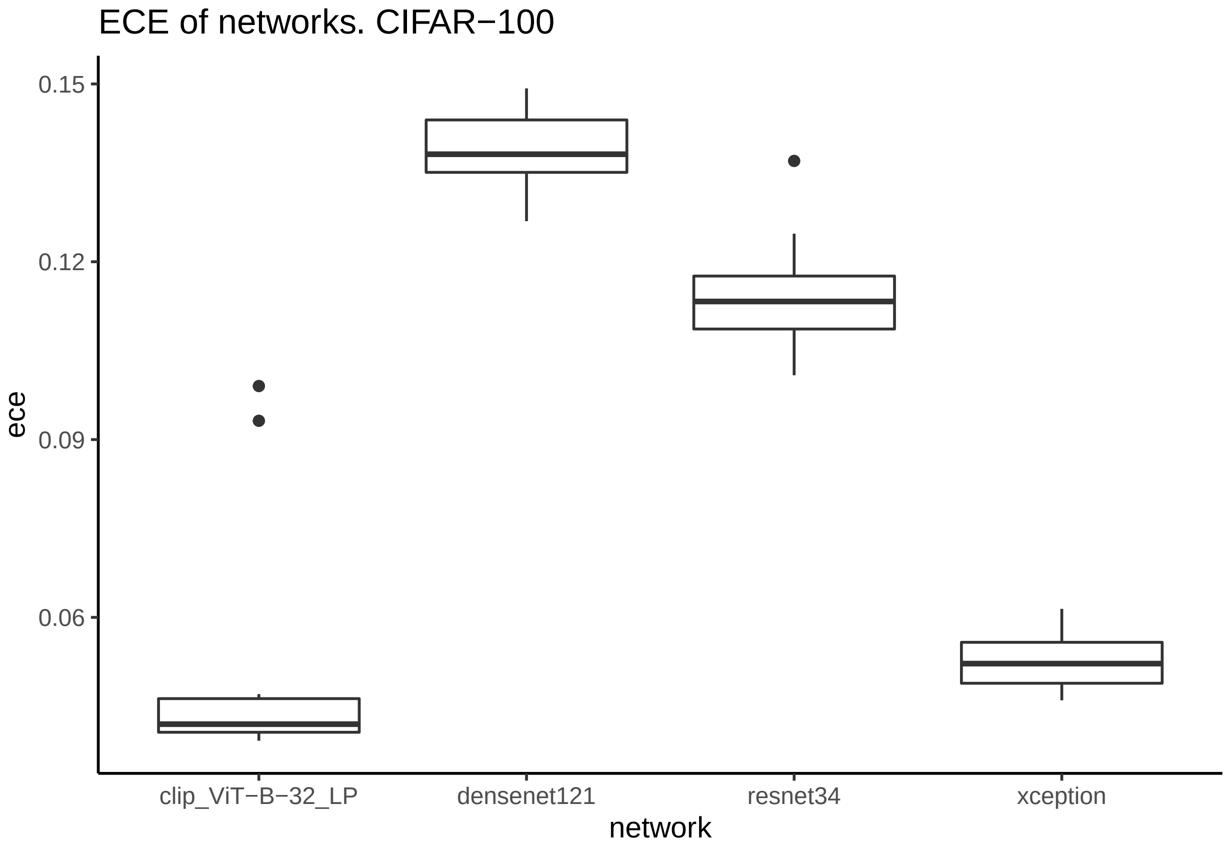


Training on the same training data as the neural networks were trained on in most cases seems to provide worse accuracy, worse nll and worse ece compared to training on separate validation set.

```r
for (met_i in seq_along(metrics))
{
    box_net <- net_results_c100 %>% ggplot() +
        geom_boxplot(mapping=aes_string(x = "network", y = metrics[met_i])) +
        ggtitle(paste0(metric_names[met_i], " of networks. CIFAR-100")) +
        theme_classic()
    print(box_net)
}
```


accuracy of networks. CIFAR−100
accuracy
0.775
0.750
0.725
0.700
network
clip_ViT−B−32_LP
densenet121
resnet34
xception

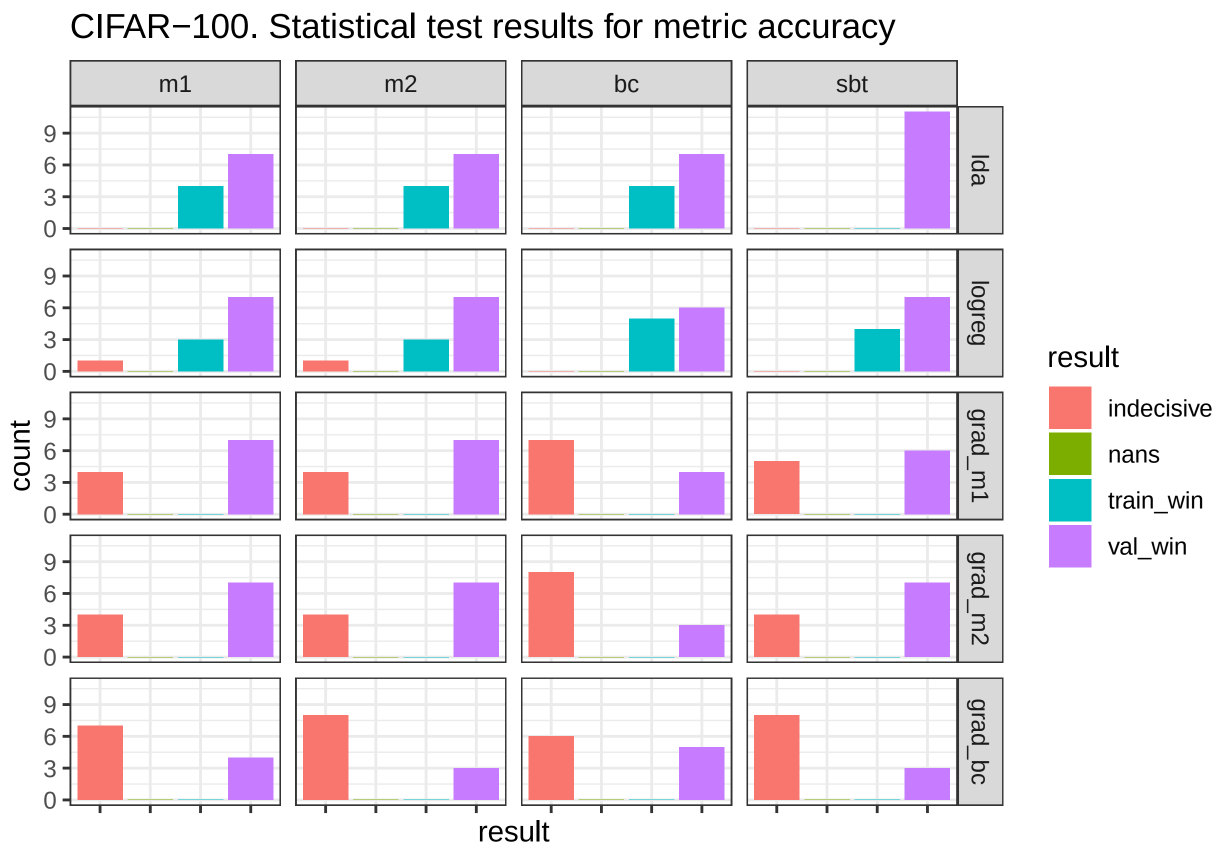

NLL of networks. CIFAR−100
nll
1.2
1.0
0.8
clip_ViT−B−32_LP
densenet121
resnet34
xception
network

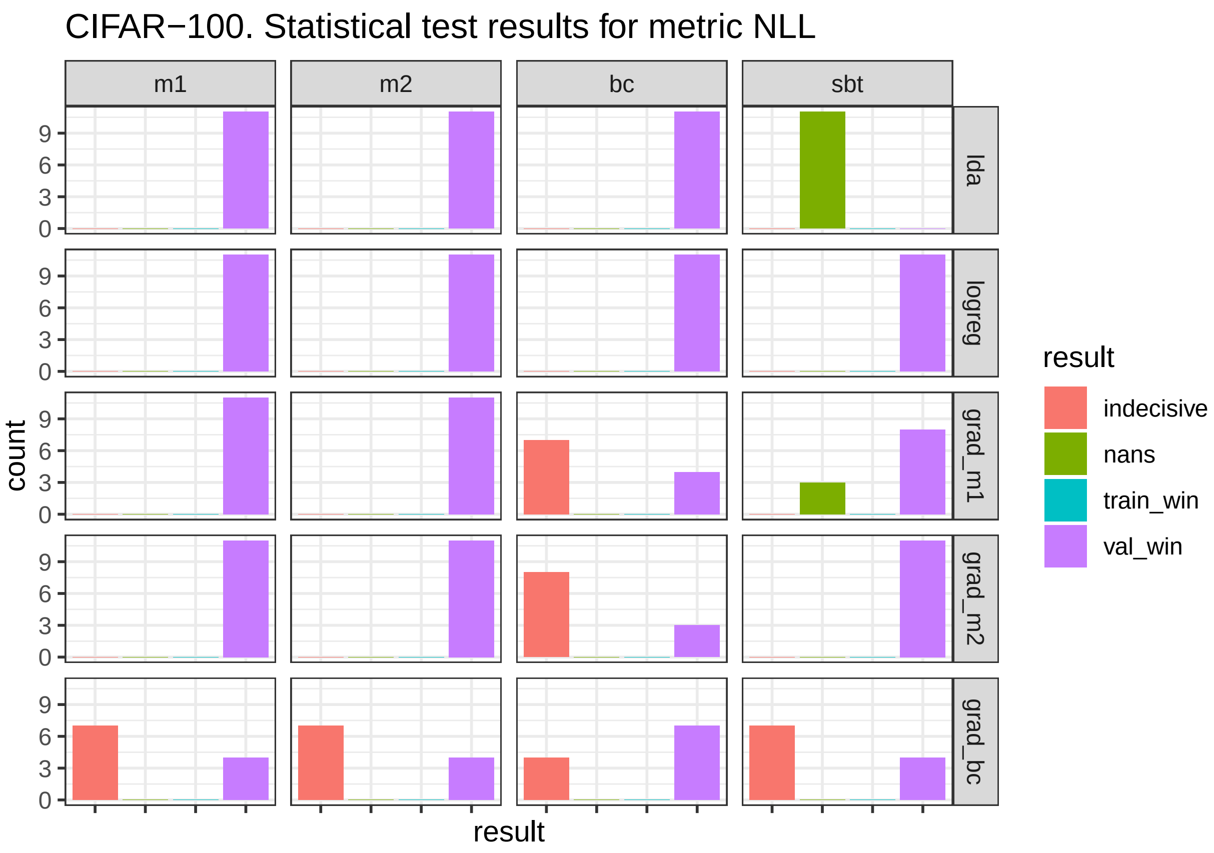


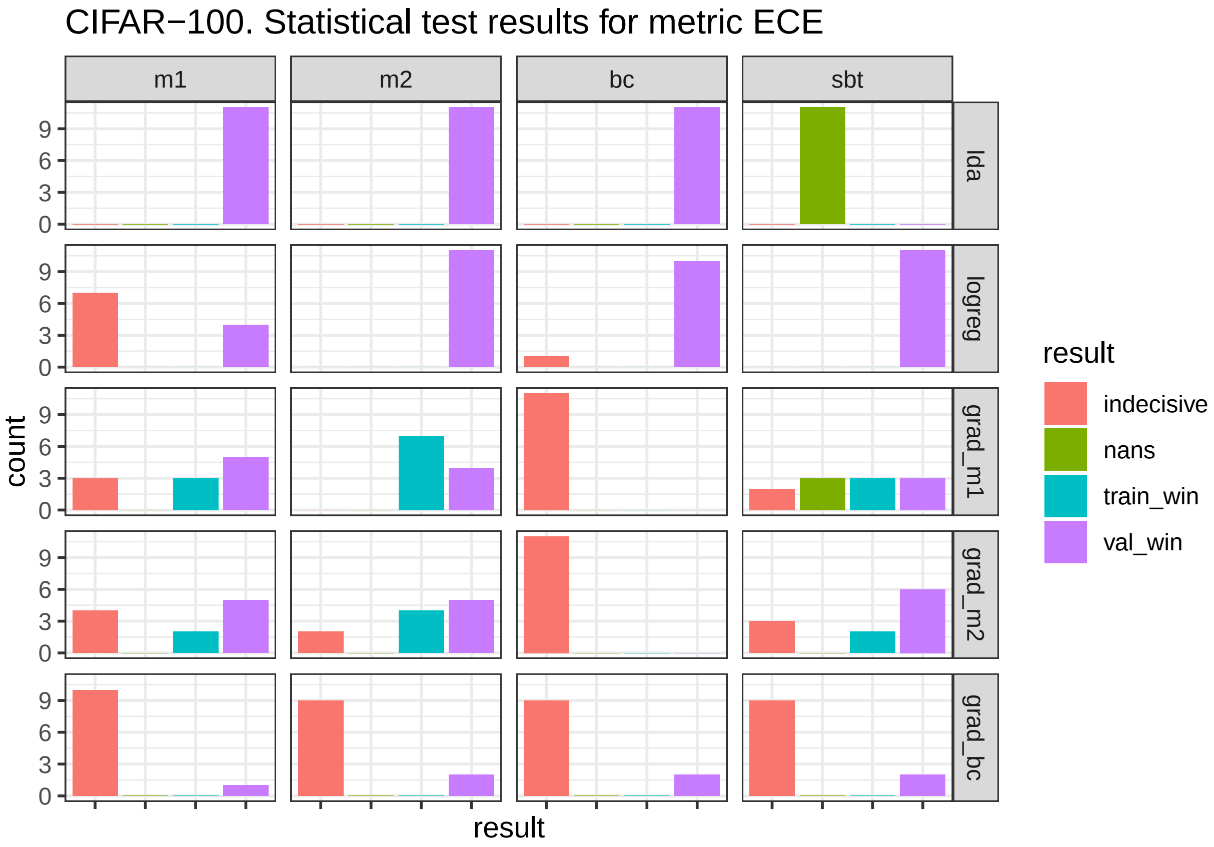
Testing statistical significance of difference for training on validation and train data. We have 50 samples from each distribution, so we will suppose the distributions are normal.

```
tests_df = expand.grid(
    combining_method = unique(ens_results_c100$combining_method),
    coupling_method = unique(ens_results_c100$coupling_method),
    metric = metrics,
    val_win = c(0),
    train_win = c(0),
    indecisive = c(0),
    nans = c(0))
sig_l <- 0.01

for (co_m in unique(ens_results_c100$combining_method))
{
    cur_co_m <- ens_results_c100 %>% filter(combining_method == co_m)
    for (met_i in seq_along(metrics))
    {
        for (cp_m in unique(cur_co_m$coupling_method))
        {
            test_res <- list(val_win = 0, train_win = 0, indecisive = 0, nans = 0)
            for (comb_id in unique(cur_co_m$combination_id))
            {
                cur_co_m_cp_m <- cur_co_m %>% filter(coupling_method == cp_m, combination_id == comb_id
                cur_co_m_cp_m_train <- cur_co_m_cp_m %>% filter(train_set == "tt")
                cur_co_m_cp_m_val <- cur_co_m_cp_m %>% filter(train_set == "vt")
                if (any(is.na(cur_co_m_cp_m_train[[metrics[met_i]]])) |
                    any(is.na(cur_co_m_cp_m_val[[metrics[met_i]]])))
                {
                    test_res[["nans"]] <- test_res[["nans"]] + 1
                }
                else
                {
                    testr <- t.test(cur_co_m_cp_m_train[[metrics[met_i]]], cur_co_m_cp_m_val[[metrics[me
                    if (testr$p.value >= sig_l)
                    {
                        test_res[["indecisive"]] <- test_res[["indecisive"]] + 1
                    }
                    else
                    {
                       if (
                           (metrics_opt[met_i] == "min" & testr$estimate[[1]] > testr$estimate[[2]]) |
                           (metrics_opt[met_i] == "max" & testr$estimate[[1]] < testr$estimate[[2]]))
                        {
                            test_res[["val_win"]] <- test_res[["val_win"]] + 1
                        }
                        else
                        {
                           test_res[["train_win"]] <- test_res[["train_win"]] + 1
                        }
                    }
                }
            }
            tests_df[
                which(tests_df$combining_method == co_m & tests_df$coupling_method == cp_m & tests_df$me
                c("val_win", "train_win", "indecisive", "nans")] <- test_res
        }
    }
}
```

```
tests_df_longer <- pivot_longer(data = tests_df, cols = c("val_win", "train_win", "indecisive", "nans")
for (met_i in seq_along(metrics))
{
    col_plot <- tests_df_longer %>% filter(metric == metrics[met_i]) %>% ggplot() +
        geom_col(mapping = aes(x = result, y = count, fill = result)) +
        facet_grid(rows=vars(combining_method), cols=vars(coupling_method)) +
        ggtitle(paste0("CIFAR-100. Statistical test results for metric ", metric_names[met_i])) +
        theme_bw() +
        theme(
            axis.text.x = element_blank())

    print(col_plot)
}
```


CIFAR−100. Statistical test results for metric accuracy
m1
m2
bc
sbt
lda
logreg
grad_m1
grad_m2
grad_bc
count
9
6
3
0
result
result
indecisive
nans
train_win
val_win


CIFAR−100. Statistical test results for metric NLL
m1
m2
bc
sbt
lda
logreg
grad_m1
grad_m2
grad_bc
count
0
3
6
9
result
result
indecisive
nans
train_win
val_win



The results show that ensemble training on validation data provides superrior accuracy in majority of cases and for all combining methods. Additionally, if we look at the individual cases, we can see, that improvements of validation training onver train training in cases where validation training wins tends to be bigger than improvements of train training over validation training in cases where train training is better. Similar holds for metrics NLL and ECE. For grad methods, however, the differences are smaller and more cases are indecisive.