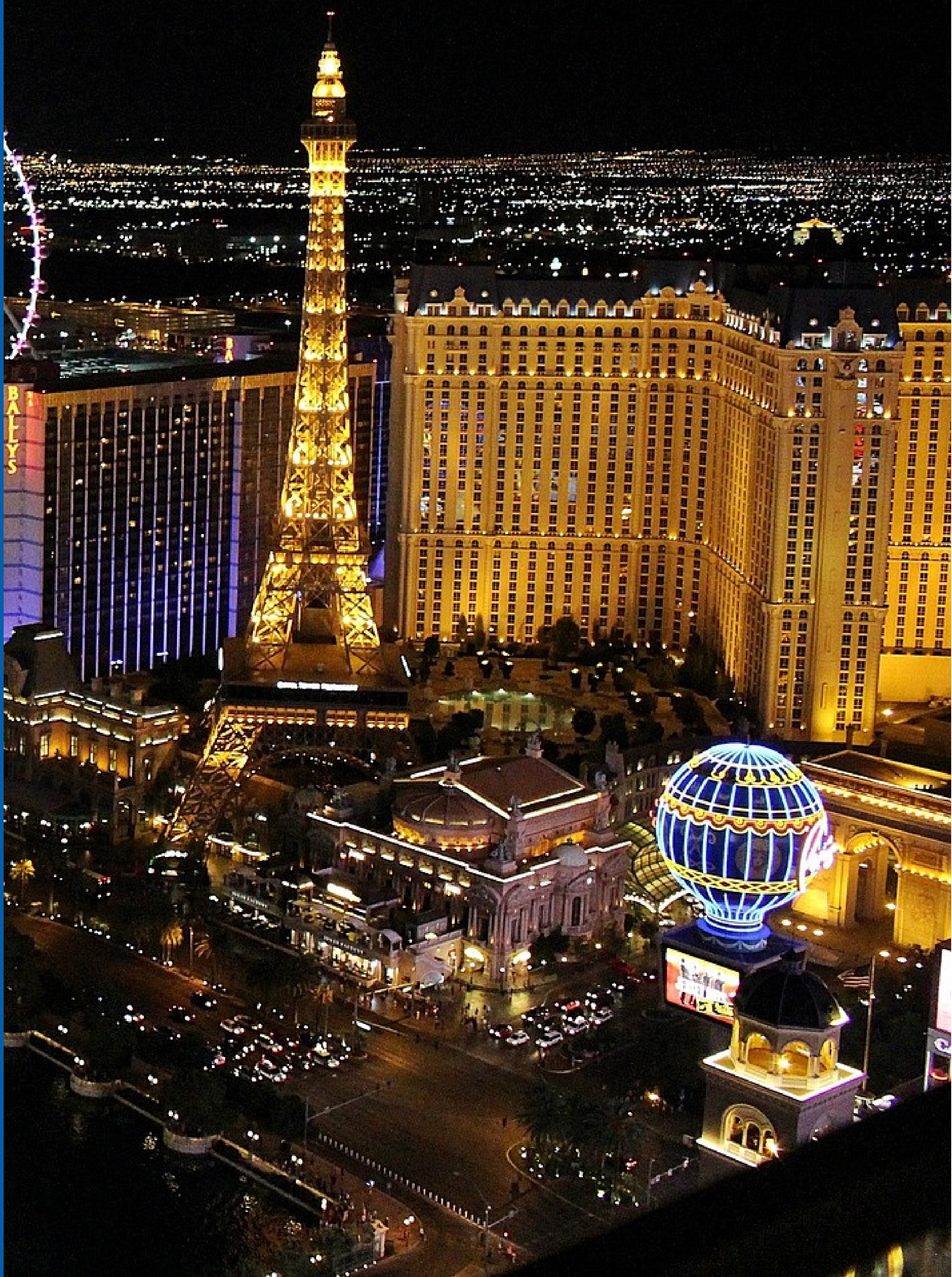
Integrating Exploratory and Simulation Modeling into Regional Transportation Planning



Prepared for

The Regional Transportation Commission of Southern Nevada

Prepared by

Caliper Corporation
1172 Beacon Street
Suite 300
Newton, MA 02461
Phone: 617-527-4700

©2021 Caliper Corporation,
All Rights Reserved

Disclaimer

The preparation of this report has been financed in part through grant[s] from the Federal Highway Administration and Federal Transit Administration, U.S. Department of Transportation, under the Metropolitan Planning Program, Section 104(f) of Title 23, U.S. Code. The contents of this report do not necessarily reflect the official views or policy of the U.S. Department of Transportation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 ACKNOWLEDGEMENTS

The project Integrating Exploratory and Simulation Modeling in Regional Transportation Planning was made possible by the cooperative efforts of the following individuals and organizations who contributed significantly to the successful completion of the project:

RTC Board of Commissioners

Debra March (Chairwoman), Mayor of Henderson

Justin Jones (Vice-Chair), Clark County Commissioner

Carolyn Goodman, Mayor of Las Vegas

Tick Segerblom, Clark County Commissioner

Isaac Barron, North Las Vegas Councilman

Stavros S. Anthony, Las Vegas Mayor Pro Tem

George Gault, Mesquite Councilman

Claudia Bridges, Boulder City Mayor Pro Tem

RTC Executive Management

M.J. Maynard, Chief Executive Officer

David Swallow, Deputy Chief Executive Officer

Francis Julien, Deputy Chief Executive Officer

Angela Castro, Chief Strategy, Policy & Marketing Officer

Marc Traasdahl, Chief Financial Officer

FHWA Travel Model Improvement Program (TMIP)

Sarah Sun, Outreach Manger

Nevada Department of Transportation

William Story, Assistant-Chief, Multimodal Planning/Program Development

Mark Wooster, Assistant Chief Traffic Information Systems Assistant Chief

RTC Management and Staff

Beth Xie, Manager of Planning (Project Manager)

Chin-Cheng Chen, Principal Transportation Planner

Hui Shen Senior Transportation Planner

Lijuan Su, Senior Transportation Planner

Grant Shirts, Transportation Planner

Gang Xie, Senior Project Engineer (FAST)

Joanna Wadsworth, Manager of Engineering (FAST)

Deborah Reardon Manager of Planning

Caliper Consultant Staff

Daniel Morgan, Vice President, Traffic Simulation

Ramachandran Balakrishna, Principal Transportation Scientist

Arun Akinnepally, Senior Transportation Scientist

Andres Rabinowicz, Vice President, Transportation Software

Jim Lam, Director, TransCAD Applications and Support

Howard Slavin, President

2.0 EXECUTIVE SUMMARY

This report describes an effort to integrate two emerging transportation planning tools – dynamic traffic assignment (DTA) and exploratory modeling and analysis (EMA) – to begin to develop practical knowledge around how these tools might be used together to improve decision-making.

On their own, the benefits of DTA and EMA are clear and have been demonstrated in prior research and in current practice. DTA enables planners to evaluate projects with greater operational sensitivity and increased temporal resolution than its static traffic assignment counterpart. Static assignments are the traditional tool of planners for analyzing congestion patterns, but they, unlike DTA, do not account for the effects of time-varying demand, queuing, and queue spillback.

EMA, for its part, gives planners a roadmap for grappling with deep uncertainty in the decision- and policy-making process. Deep uncertainty is uncertainty surrounding one or more variables whose influence in forecasting models is critical yet whose values cannot be chosen with confidence, consensus, or with the benefit of empirical evidence. EMA leverages existing models, such as forecasting models currently used by the RTC, to explore an array of possible outcomes and from them to discern patterns or trends that will help inform decision-making.

For example, it can be difficult to plan for the shifts in regional traffic patterns that are likely to occur if high-speed rail between Los Angeles and Las Vegas is constructed and becomes fully operational. How many trips will arrive via high-speed rail that would have otherwise arrived by car or via Harry Reid International Airport? What will happen to fuel prices and how will they change the visitor mode choice calculus? Moreover, if connected and autonomous vehicles (CAVs) grow in acceptance and market share, will a trip by autonomous vehicle become more competitive with high-speed rail service? Uncertainty is intrinsic to forecasting in any discipline, but to answer questions such as these requires that certain assumptions be made, assumptions that cannot be made with confidence or around which there is difficulty building consensus.

EMA does not make one assumption but explores many drawn from a specified distribution. Moreover, EMA provides a means of analyzing one or more policy strategies for mitigating the adverse impacts of the uncertainty variables. These policy levers may be effective only in isolation, or they may be effective only in combination with one another, or they may be effective only across certain ranges of value on the uncertain variables. The combinatory effects of the set of factors under consideration – uncertainty variables and policy levers – stress the limits of conventional scenario planning approaches and underscore the advantage of EMA.

EMA provides a methodology for:

1. organizing the assumed values for any number of analysis factors, uncertainty variables and policy levers included, into inputs to a core model,
2. executing the model experiments by invoking that core model, and
3. collating output performance measures produced by the model for visualization and analysis.

These various functions are operationalized in an open-source software toolbox developed prior to this effort with the stewardship and funding of the Federal Highway Administration's (FHWA) Travel Model Improvement Program (TMIP). This toolbox is called the TMIP EMA Toolbox, or TMIP-EMAT.

A growing body of research and some practical applications have established the promise of integrating TMIP-EMAT with travel demand forecasting models. The effort described in this report treads new ground by integrating TMIP-EMAT with DTA. The DTA models that were integrated with TMIP-EMAT are those at RTC's disposal. One of the work products of the effort is a practical software application that planners at RTC can leverage to perform their own EMA in the future. The other work products include demonstration and documentation of the application to explore the impacts of increased traffic drawn to T-Mobile Arena for a Las Vegas Golden Knights (VGK) game and several congestion mitigation measures.

This report describes the application development effort, summarizes the demonstration of the application with the VGK scenario, and presents conclusions and lessons learned. The conclusions and lessons learned are intended both to provide guidance to the RTC in its adoption and deployment of the application and to provide substantive contribution to EMA research for the benefit of the RTC's peers in the broader transportation planning profession.

3.0 A PRACTICAL EMAT APPLICATION

Integration with TransCAD and TransModeler

This section describes the integration of the EMAT application with core models in TransCAD and TransModeler. The objective of the integration is to allow EMAT to interface with the RTC travel demand model (TDM) and TransDNA mesoscopic dynamic traffic assignment (DTA) model in TransCAD and with the microscopic DTA model in TransModeler. More information and background about EMAT itself are available at https://tmip-emat.github.io/index.html.

The application uses an EMAT class implemented in the GIS Developer's Kit (GISDK), an application programming interface (API) shared by TransCAD and TransModeler. For more information about GISDK and about the EMAT class, refer to Appendix A: User's Guide. The EMAT class manages EMAT executions of the core model (e.g., the TDM or the DTA) in TransCAD or TransModeler.

A scripting language is one element of the GISDK. Script is commonly written in the GISDK to develop custom applications such as the RTC TDM for TransCAD. In GISDK script, an instance of the EMAT class is created as follows:

```
obj = CreateObject("EMAT", {Enviroment : "emat"})
```

To describe this command, some terminology that is not necessarily commonplace in the transportation planning vernacular must first be defined. Before the EMAT application described in this report can be used with TransCAD and TransModeler, EMAT must first be properly set up on the computer. EMAT is not installed like a typical Windows application such as TransCAD or TransModeler, which are installed on your computer by an installer. Rather, EMAT is created as a Conda environment, where:

- Conda is a package manager for Python users,
- Python is a popular programming language among data scientists and transportation modelers,
- packages are collections of software tools and utilities, and
- an environment is a directory into which a set of packages is installed so that they can be called upon when that environment such as the EMAT environment is in use.

Anaconda, a software distribution that bundles Conda with a large number of packages that are useful in a broad range of data science applications, is recommended by TMIP-EMAT's creators. Anaconda is installed like a typical Windows application and can be freely downloaded from Anaconda's website. More information about Anaconda and Python in relation to EMAT and links to download Anaconda are found in the EMAT documentation online: https://tmip-emat.github.io/source/emat.conda.html.

In the command above, the second argument – `{Enviroment : "emat"}` – specifies the name given to the Conda environment in which the packages required by EMAT are installed. This is the Conda environment that is activated by TransCAD or TransModeler when the EMAT application is invoked. The link above contains helpful information about creating an appropriate environment for EMAT.

The variable **`obj`** is an object instantiated from the EMAT class and is the handle through which EMAT methods can be accessed. For example, a scope file is added with the **`AddScopeFile()`** method:

```
obj.AddScopeFile({Filename: scopeFile})
```

The scope file is in yaml format as required by EMAT and lays out the parameters of the experimental design, including the uncertainty variables and performance measures. Please see the EMAT documentation for the exact format of the scope file: https://tmip-emat.github.io/source/emat.scope/scope.file.html.

After creating the EMAT object and adding a scope file to it, the next step is to generate experiments:

```
opts = {
        addMetricFields: True,
        n_samples_per_factor: 20,
        sampler: "lhs"
}
obj.CreateExperiments({
        DesignFileName: experimentFile,
        Args: opts
})
```

The **`CreateExperiments()`** method will invoke the EMAT method **`design_experiments()`**. The **`CreateExperiment()`** method accepts all the arguments that are expected by design_experiments() in EMAT. The arguments to **`design_experiments()`** must be passed using the Args keyword as shown above. The **`CreateExperiment()`** method will produce a table with all the experiment specifications, including the values of the input parameters, that will be used to run each core model experiment and produce the corresponding output performance measures (Figure 1).

Dataview1 - rtc_design_v2

Experiment	DeliveriesPerTour	ECommerceDiversion	ECommerceTruckSplit	HSR_Pass	IVPH_Pass	MIA_Pass	PackagesPerTrip
1	109.5830	0.1113	0.3973	19978.0827	17798	227646.5857	5.2147
2	238.2812	0.1997	0.4176	18401.1854	18788	173756.3322	4.0231
3	230.7968	0.2729	0.4125	19669.5032	14797	206045.8730	4.0600
4	235.6655	0.1399	0.4874	22677.0068	13293	152361.3262	7.1564
5	141.6388	0.1707	0.4114	22811.6934	16032	166146.1271	6.4448
6	151.4709	0.1460	0.4186	20348.9622	13056	148497.7544	7.0439
7	206.5627	0.2327	0.4144	20698.5282	14504	165594.2106	4.9196
8	138.7344	0.1917	0.3938	19889.6752	15947	135341.4990	7.1311
9	151.3583	0.2359	0.4669	21925.0039	15346	110472.4094	4.9653
10	160.4653	0.1296	0.4384	19431.6611	11438	134417.0142	4.6285
11	233.1337	0.1414	0.4308	18081.3996	9596	171595.3719	6.8586
12	177.7969	0.1422	0.3813	21732.2050	16995	157188.4757	5.4298
13	297.0291	0.2563	0.3410	18693.4773	13387	147764.6844	4.3865
14	136.9701	0.1057	0.4361	20959.3142	14057	190601.3869	6.1601
15	258.9323	0.1209	0.3545	18870.0548	11635	151066.9817	4.4595
16	161.9257	0.1942	0.3069	19551.0079	11158	168848.6089	7.9060
17	157.4037	0.2442	0.4992	23700.3688	18165	195294.5245	7.6319
18	252.7169	0.1956	0.4334	18134.0367	18447	128209.9434	7.6056
19	229.5808	0.1117	0.3479	19076.2656	13919	173771.5743	6.2794
20	265.3238	0.2037	0.3440	21417.0178	8841	117958.5423	6.7559
21	232.7144	0.2620	0.4410	23924.5446	15806	131248.6488	4.2088
22	256.8274	0.2148	0.3923	19881.4959	18617	164447.8353	7.5819

Figure 1: Sample experiment table

To run the experiments, the EMAT class will interface with the core model and a metrics function. Specifically, the EMAT class calls upon a particular scenario defined in the flowchart model in which the core model is implemented. The flowchart model will be used to create child scenarios representing each experiment. This ensures that the original flowchart model is not unduly changed by the EMAT application. The metrics function post-processes outputs from the model run and returns the performance measures as an options array, where an options array is simply an array of name-value pairs (e.g., name “Filename” and value scopeFile, where scopeFile is a string variable indicating the file path and name of the scope file). The GISDK script commands to run the EMAT experiments are as follows:

```
modelOpts = {
        Name: modelFile,
        ScenarioName: scenarioName,
        StepName: stepNameToRun
}
measureOpts = {
        Function: “retrieveMeasures”,
        UI: measuresFunctionUI
}
obj.RunExperiments({
        Model: modelOpts,
        ExperimentTable: experimentFile,
        Measures: measureOpts
})
```

In GISDK, script files, called resource files, are compiled into a user interface (UI) database. A TDM in TransCAD, for example, is typically compiled into such a UI database. The metrics function can be compiled in a UI database separately from the one in which the core model might be compiled (e.g., option name UI and value `measuresFunctionUI` above).

When the `RunExperiments()` method is invoked, the application will then run the model in a loop (over the records in the experiment table) and populate the table with the output metrics for each experiment. After the experiment runs are finished, the outputs can be visualized, and EMAT analysis can be carried out. This is detailed in the next section.

A Graphical User Interface for EMAT in TransCAD and TransModeler

With the application that was developed in TransCAD and TransModeler to interface with EMAT, it is possible to design and run EMA experiments with the TDM and DTA core models without ever writing script that invokes the EMAT class described above. This is because a graphical user interface (GUI) will perform the work of invoking EMAT and the core model experiments given a scope file and other inputs.

The EMAT application GUI in TransCAD and TransModeler begins with an EMAT toolbar with the title “EMAT Tools” (Figure 2).

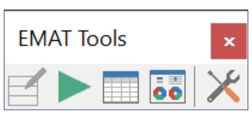


Figure 2. The EMAT application toolbar in TransCAD and TransModeler

The buttons on the toolbar perform a variety of tasks in the EMA process (Table 1).

Table 1. Buttons in the EMAT application toolbar in TransCAD and TransModeler

Button	Name	TASK
	Create Experiment Table	Creates a table with a record for every core model experiment that will be run, where an experiment is a combination of values of each uncertainty variable and policy lever, sampled from the relevant distributions, as defined in the scope file. The experiment table will have an empty field for each performance measure identified in the scope file to be filled when the experiments are run.
	Run Experiments	Launches the core model in series, once for each record in the experiment table, adjusting the core model parameters and inputs using the values in the experiment table and filling the measures fields from the resulting model outputs.
	Open Experiment Table	Opens the experiment table in a dataview window for review/inspection.
	Visualize Metrics	Opens a dialog box allowing several visualizations to be created from the output measures generated from the experiments (see Visualization Support for EMAT).
	Settings	Opens the EMAT Setting dialog box where the EMAT environment name, scope file, and other inputs defining the experiment design can be entered.

When the Settings button is clicked on the EMAT Tools toolbar, TransCAD or TransModeler will show the EMAT Settings dialog box (Figure 3). Before any of the EMA tasks in the toolbar can be performed, the following inputs must be provided in the EMAT Settings dialog box:

1. Conda Environment: This is the name given to the EMAT environment when it was created. Typically, this would be the string "EMAT."
2. Core Model File Name: The model file (.model) of the TDM, TransDNA map (.map) or TransModeler simulation project (.smp) with which the experiments will be run.
3. Scenario: The name of the scenario in the TDM, TransDNA Scenario Manager, or TransModeler Project Settings in which the experiments will be performed.
4. Run: If the Core Model File Name is that of a TDM (.model), then the model may be run beginning from a particular step, or the entire model may be run.
5. Scope File: The file name of the EMAT scope file (.yaml) defining the uncertainty variables, policy levers, and performance measures.
6. Experiment Table: The file name of the experiment table (.bin). This may be the name of an existing file if the experiment table was already created, or it may be the desired file name of the experiment table to be created by clicking the Create Experiment Table on the application toolbar.
7. Number of Samples: This is the value of the EMAT parameter `n_samples_per_factor` described in the EMAT documentation as "the number of samples to draw per input factor in the design." A value of 10 or greater is recommended to support meta-model estimation. The higher the value, the greater the number of experiments that will be created in the experiment table, and hence the greater the number of core model runs.

Additionally, the following inputs may be provided but are optional:

1. Conda Path: If you installed Anaconda according to the instructions provided in Appendix A, then the Conda Path can be left blank. However, in other installation scenarios, it may be necessary to provide the file path and name of the conda batch (.bat) or executable (.exe) file that should be used.
2. UI Database: The UI database in which the Parameters Macro and Measures Macro are compiled.
3. Parameters Macro: Whereas it may be possible in many instances to design an experiment and perform EMA without writing script that interacts directly with EMAT via the EMAT class described earlier in this report, it will be necessary in most, if not all, experiments, to write script that will translate parameters representing uncertainty variables in the EMAT scope file to their corresponding adjustments in core model parameters or to translate variables representing policy levers in the EMAT scope file to their corresponding changes to core model inputs.
4. Measures Macro: Similarly, if the TDM or DTA does not produce a desired performance measure by default, a macro may need to be written to summarize or aggregate the measure from the core model's standard outputs. This macro will be called after each core model run to retrieve the measure.

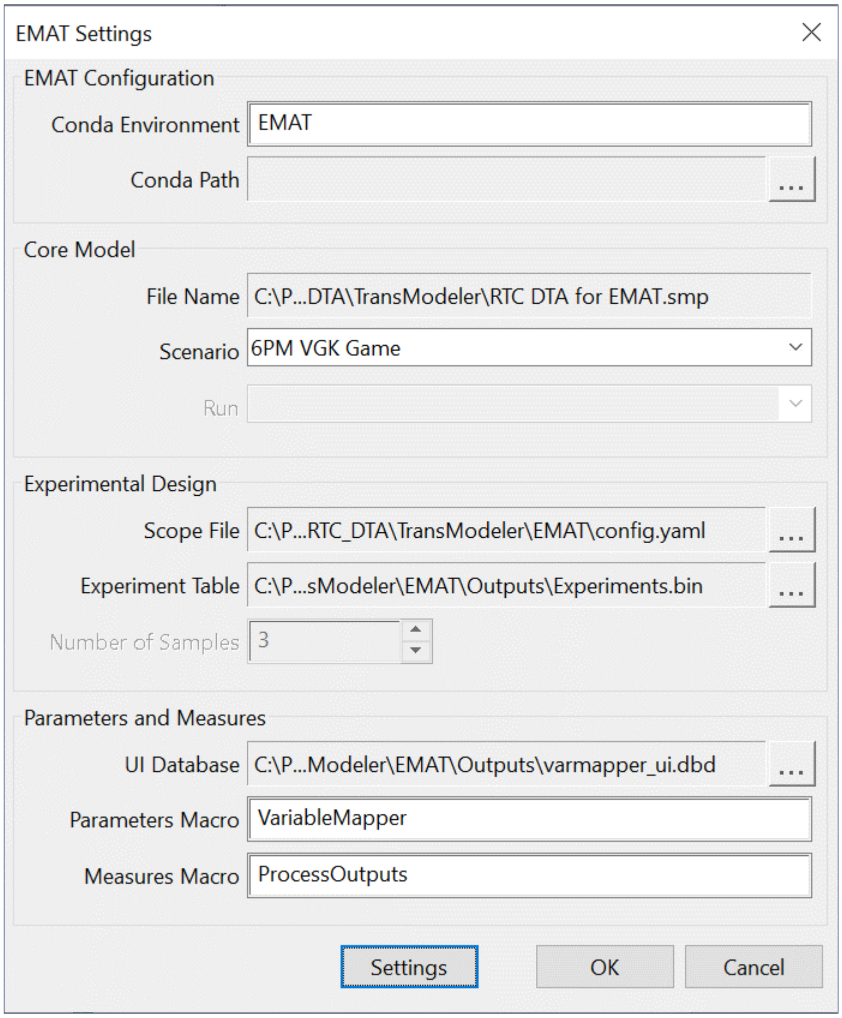


Figure 3. The EMAT Settings dialog box where the EMAT configuration and experimental design inputs are specified

Visualization Support for EMAT

Just as a GUI was developed to facilitate the preparation of the EMA inputs, a GUI was developed in TransCAD and TransModeler to support visualization of the output measures without requiring a Jupyter Notebook. Jupyter is a web-based data science platform for Python and is the primary visualization platform for prior EMAT projects performed by others. The GUI (Figure 4) is opened by clicking the Visualize Metrics button on the EMAT application toolbar.

The visualizations include the following:

1. Feature scores
2. Scatter plots
3. Threshold feature scores

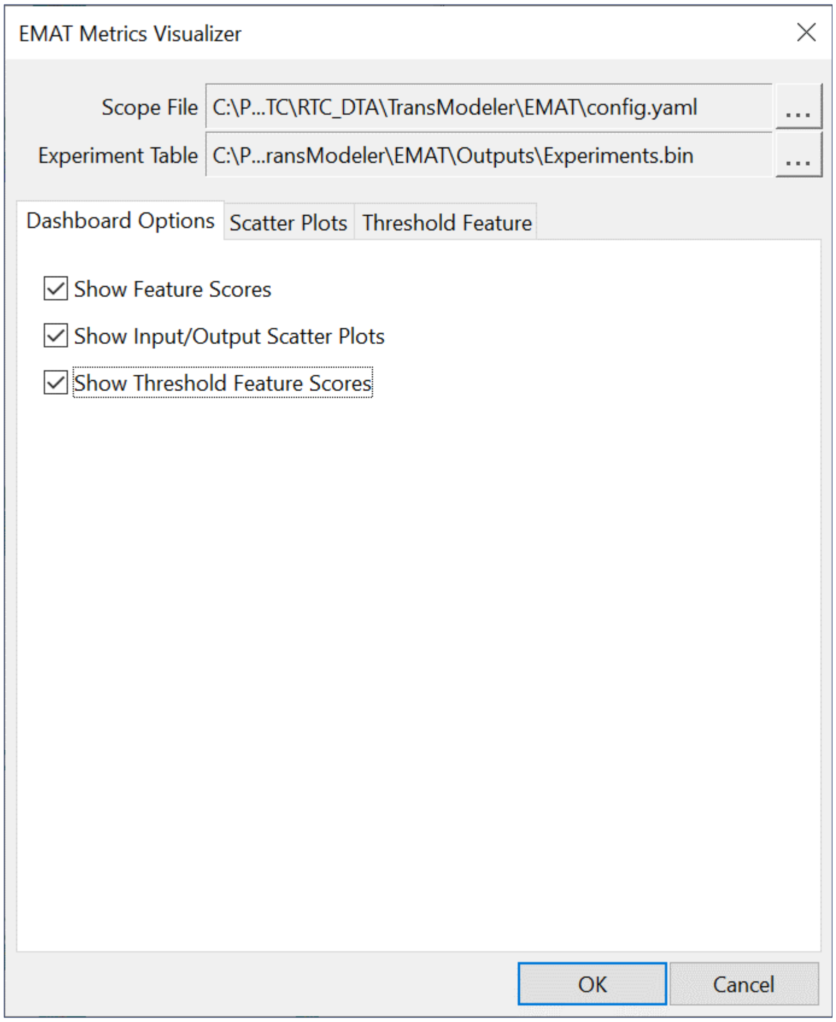


Figure 4. The EMAT Metrics Visualizer GUI

Feature scores identify the variables that have the highest influence on the output measure of interest. In other words, they identify the variables that explain the distribution of the output measure. In EMAT, the feature scores are normalized so that they add up to 1. Therefore, although the feature scores identify the influential variables, they do not indicate if the total influence of all the variables is significant.

Scatter plots of performance measures against the uncertainty variables showcase the relationships, if any, that might exist between the inputs and measures. While the scatter plots may show trends of varying shapes (e.g., linear) and suggest correlation, it is important to note that the bivariate plots do not indicate cause-effect relationships with certainty.

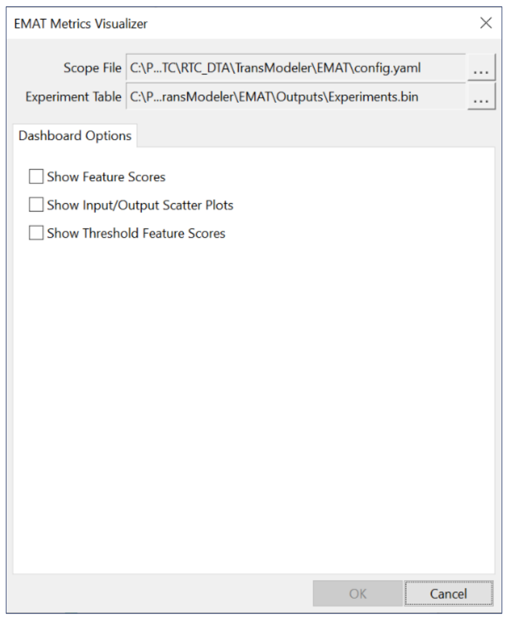
Threshold feature scores, like feature scores, identify the variables that influence an output measure; however, they focus on whether the output measure is above or below a threshold. For example, threshold feature scores identify the uncertainties that make VMT increase over 100,000. Threshold feature scores are presented for a single output measure at a time for multiple threshold values.

To invoke the dialog box requires two preconditions:

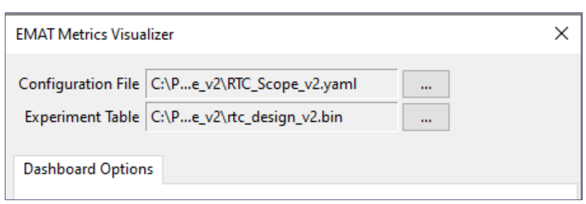
1. EMAT must be installed in one of Anaconda's Python environments, and
2. The fully populated experiment table must be available from previous experiments run with the core model.

The following steps describe how to use the application to produce performance measure visualizations:

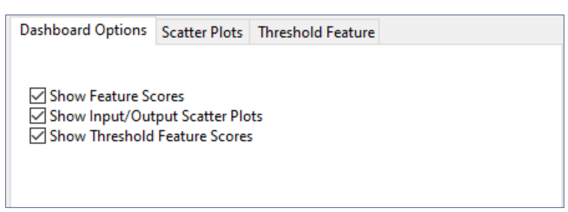
1. Open the *EMAT Metrics Visualizer* dialog box by specifying the path to Conda command and the name of the Conda environment where EMAT is installed.



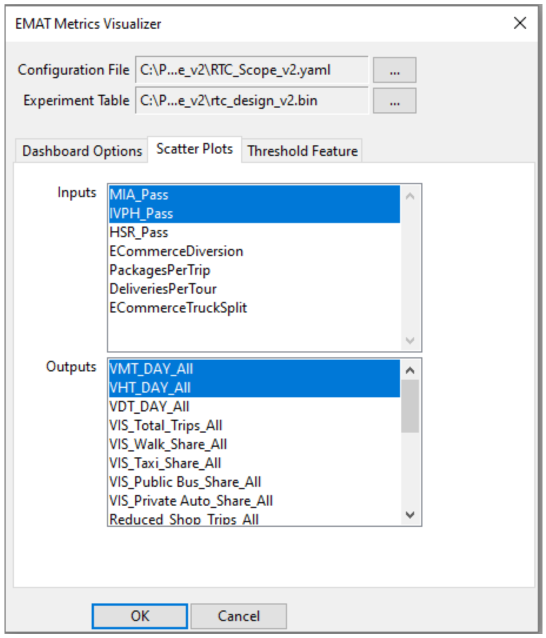
2. Click … to choose the configuration file (i.e., the scope file in yaml format), and click … to choose the experiment table.



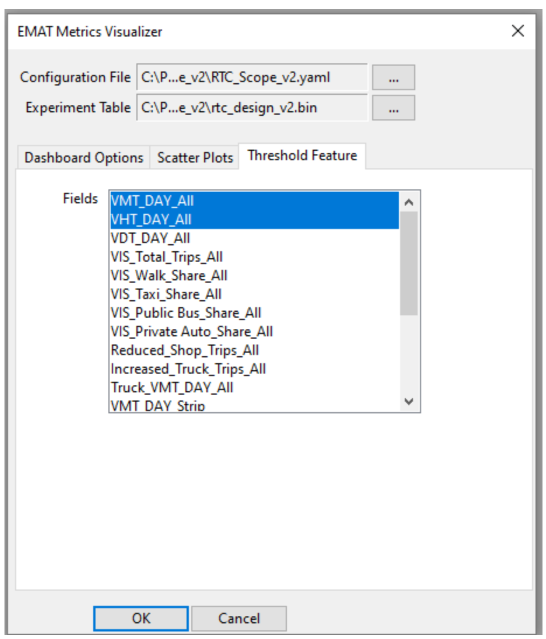
3. Check the boxes on the tab *Dashboard Options* tab next to the visualizations you wish to create. When you check the boxes, corresponding tabs appear.



4. If you checked the *Show Input/Output Scatter Plots* box, then click the *Scatter Plots* tab and choose the set of input variables and output measures to be plotted.



5. If you checked the *Show Threshold Feature Scores* box, then click the *Threshold Feature* tab and choose the set of output measures to be included among threshold feature scores.



6. Click *OK*. TransCAD will construct a dashboard and populate it with the results. Each tab in the dashboard will contain different results base on the choices that were made in the previous steps. Figure 5, Figure 6, and Figure 7 illustrate of the kinds of visualizations that are produced in the dashboard.

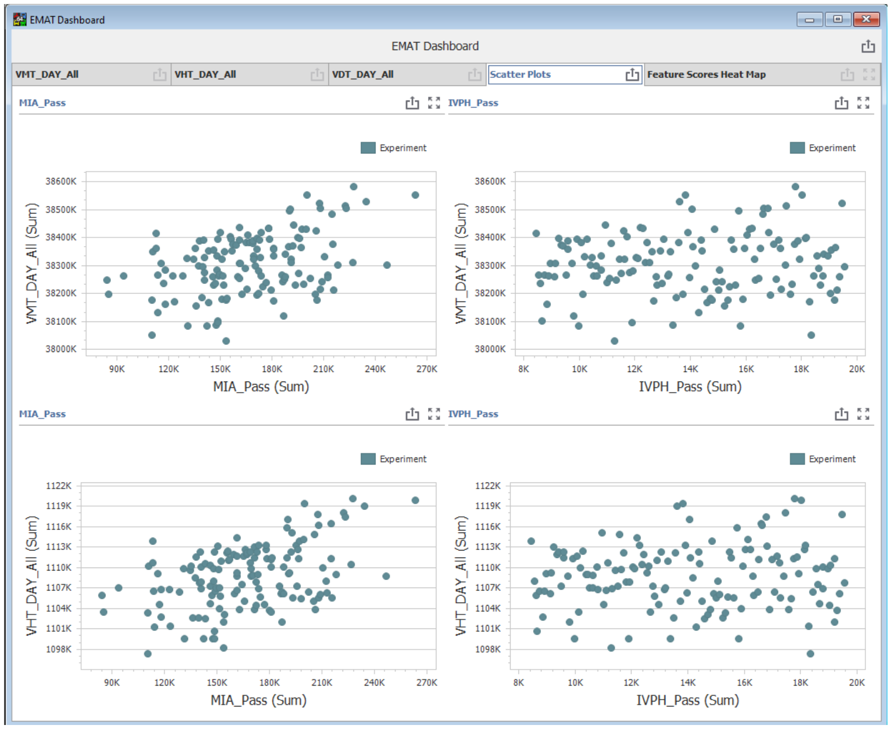
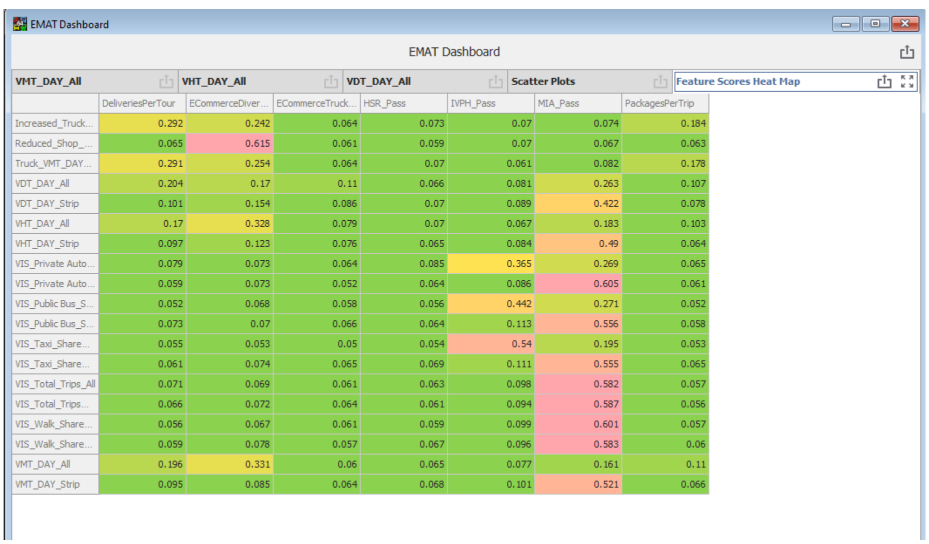


Figure 5. Scatter Plots in the EMAT Dashboard



	DeliveriesPerTour	ECommerceDiver...	ECommerceTruck...	HSR_Pass	IVPH_Pass	MIA_Pass	PackagesPerTrip
Increased_Truck...	0.292	0.242	0.064	0.073	0.07	0.074	0.184
Reduced_Shop_...	0.065	0.615	0.061	0.059	0.07	0.067	0.063
Truck_VMT_DAY...	0.291	0.254	0.064	0.07	0.061	0.082	0.178
VDT_DAY_All	0.204	0.17	0.11	0.066	0.081	0.263	0.107
VDT_DAY_Strip	0.101	0.154	0.086	0.07	0.089	0.422	0.078
VHT_DAY_All	0.17	0.328	0.079	0.07	0.067	0.183	0.103
VHT_DAY_Strip	0.097	0.123	0.076	0.065	0.084	0.49	0.064
VIS_Private Auto...	0.079	0.073	0.064	0.085	0.365	0.269	0.065
VIS_Private Auto...	0.059	0.073	0.052	0.064	0.086	0.605	0.061
VIS_Public Bus_S...	0.052	0.068	0.058	0.056	0.442	0.271	0.052
VIS_Public Bus_S...	0.073	0.07	0.066	0.064	0.113	0.556	0.058
VIS_Taxi_Share...	0.055	0.053	0.05	0.054	0.54	0.195	0.053
VIS_Taxi_Share...	0.061	0.074	0.065	0.069	0.111	0.555	0.065
VIS_Total_Trips_All	0.071	0.069	0.061	0.063	0.098	0.582	0.057
VIS_Total_Trips...	0.066	0.072	0.064	0.061	0.094	0.587	0.056
VIS_Walk_Share...	0.056	0.067	0.061	0.059	0.099	0.601	0.057
VIS_Walk_Share...	0.059	0.078	0.057	0.067	0.096	0.583	0.06
VMT_DAY_All	0.196	0.331	0.06	0.065	0.077	0.161	0.11
VMT_DAY_Strip	0.095	0.085	0.064	0.068	0.101	0.521	0.066

Figure 6. Feature Scores in the EMAT Dashboard

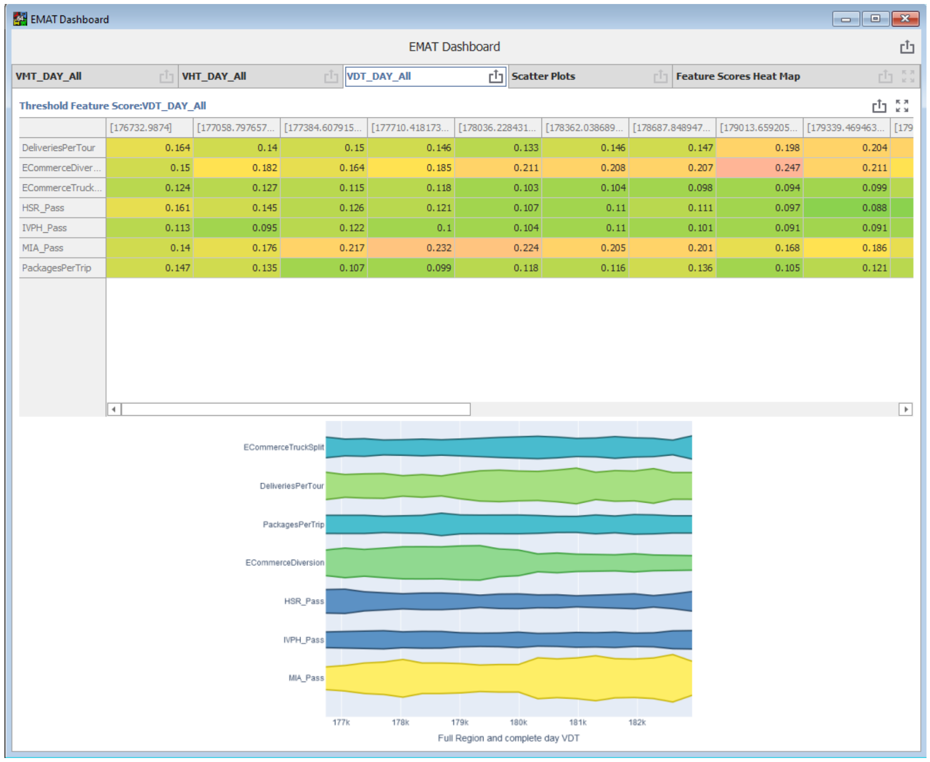


Figure 7. Threshold Feature Scores in the EMAT Dashboard

4.0 EXPLORATORY ANALYSIS WITH THE REGIONAL TRAVEL DEMAND MODEL

The RTC Travel Demand Model as Core Model

The EMAT application described in the previous section is designed to interface with a model in the flowchart interface in TransCAD. The RTC TDM was migrated from its prior interface to the flowchart interface as part of this project. The flowchart interface has much of the same core functionality as the previous-generation interface but has a more modern and flexible implementation that modularizes steps of the model workflow more formally and provides more efficient management of scenarios and the changes to model inputs and parameters that distinguish one scenario from another.

Scenarios in the flowchart interface may represent different forecast years and different assumptions about which projects will be constructed or policies adopted. Importantly for EMAT, the scenarios also allow for different model variables and parameters to be changed. Changes can be made in child scenarios of a parent scenario so that the model's original scenarios are not changed. Further, only those inputs that are changed are saved as part of a child scenario. The rest of the scenario's inputs are adopted from the parent scenario from which the child scenario was spawned. This provides a lean, flexible framework for constructing experiments that are run with varied inputs as part of EMA.

The RTC TDM in the flowchart interface is pictured in Figure 8. Tools for choosing and managing scenarios appear in a model toolbar when the model is opened, and inputs to, outputs of, and parameters in each step of the model can be reviewed and modified interactively by clicking and right-clicking the steps displayed in the model window. The flowchart interface also represents a more intuitive visualization of the model workflow, which may include arrows representing feedback between traffic assignment and trip distribution.

More information about the flowchart model and how to manage scenarios is available in the RTCSNV TransCAD Flowchart Model User's Guide.

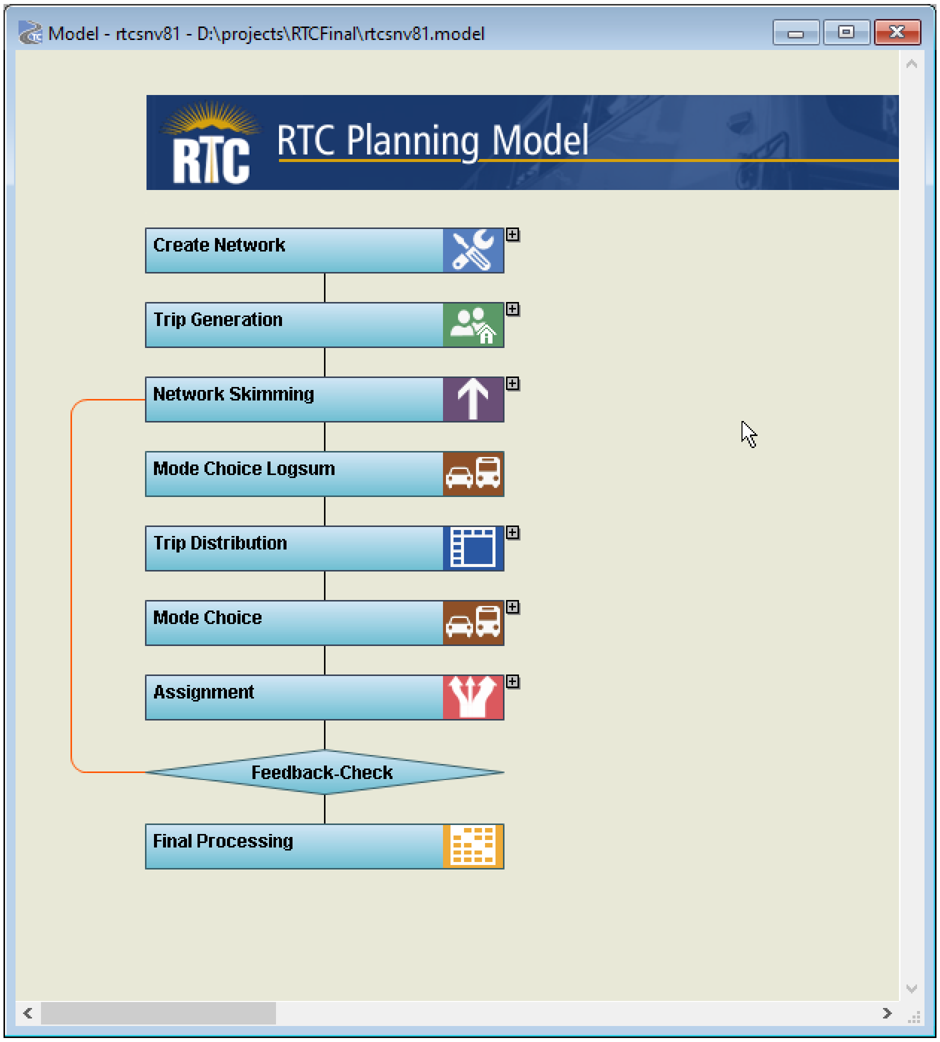


Figure 8. The RTC Travel Demand Model in the Flowchart Interface

Uncertainty Variables

Efforts were undertaken early in the project schedule to identify variables having critical influence on travel in the region but around which there is considerable uncertainty. Stakeholders were gathered in a workshop and invited to provide input. Several variables were discussed and are summarized in Appendix B. These variables were compiled into a list for consideration among the project team and RTC. Of these variables, two were determined to have representation in the travel demand model adequate to support EMA. Others would require significant updates to the TDM and are provided in Appendix B for consideration in future EMA exercises.

The two uncertainty variables chosen for analysis are:

1. Visitors: includes the uncertainties with respect to the number of visitors to the study area from airports, high-speed rail, and the I15 gateway and their effect on network performance

2. E-commerce: includes uncertainties regarding the number of shopping trips that are replaced by packages and their effect on network performance

In the EMAT framework, scenarios are constructed and analyzed that vary visitor and e-commerce activity simultaneously to elicit their combined effects on transportation system performance measures, described later in this report, from the TDM.

Visitor Uncertainty

The COVID-19 pandemic shined a bright light on the impermanence of visitor travel in the Las Vegas metropolitan area and on the need for tools such as EMAT to better understand how transportation system performance might respond when visitor traffic is impacted by forces outside of policy-makers' control and beyond the TDM's predictive powers. In the context of the TDM, four variables that together determine visitor traffic in the region were identified:

- *MIA_Pass*
- *IVPH_Pass*
- *HSR_Pass*
- *I15Gateway_Reduction*

The variables and their respective roles in the TDM are discussed below.

The variables *MIA_Pass* and *IVPH_Pass* represent the passengers arriving at the McCarran and Ivanpah Valley International Airport airports, respectively. The Ivanpah airport is a proposed relief airport for McCarran and as such does not draw traffic or employment in the TDM's base year scenarios. In the TDM, the passengers arriving at the airports are modeled as trips generated at the corresponding zones. The base year value of *MIA_Pass* is 110,025, and the base year value of *IVPH_Pass* is 0.

The *HSR_Pass* variable represents the number of high-speed rail passengers visiting the area at the proposed high-speed rail station. Its value is zero for the base year and 23,836 in the 2050 demographics table. This generates high-speed rail visitor trips, but it also interacts with the third visitor parameter *I15Gateway_Reduction* in the internal-external/external-internal (IXXI) models.

The *I15Gateway_Reduction* variable reflects the expected reduction in IX auto passengers into Las Vegas if there is any high-speed rail service. As illustrated in Figure 9, TAZ 900 is the high-speed rail station into Las Vegas, and TAZ 2506 is the I15 Gateway entry point. The logic here is that if high-speed rail exists and serves the region from points south of Las Vegas, that will reduce auto trips into Las Vegas. The *I15Gateway_Reduction* is always found to be ½ of *HSR_Pass* in the model. The assumption is that two *HSR_Pass* trips imply a reduction of one two-person car trip from I15 gateway. Therefore, this variable does not need to be explicitly considered in modeling visitor uncertainty because it already is implicitly considered.

Based on the discussion above, the uncertainty in visitor traffic is captured in the TDM by varying the values of the following variables:

1. *MIA_Pass*: Number of passengers to McCarran airport

2. *IVPH_Pass*: Number of passengers to IVPH airport
3. *HSR_Pass*: Number of passengers using high-speed rail

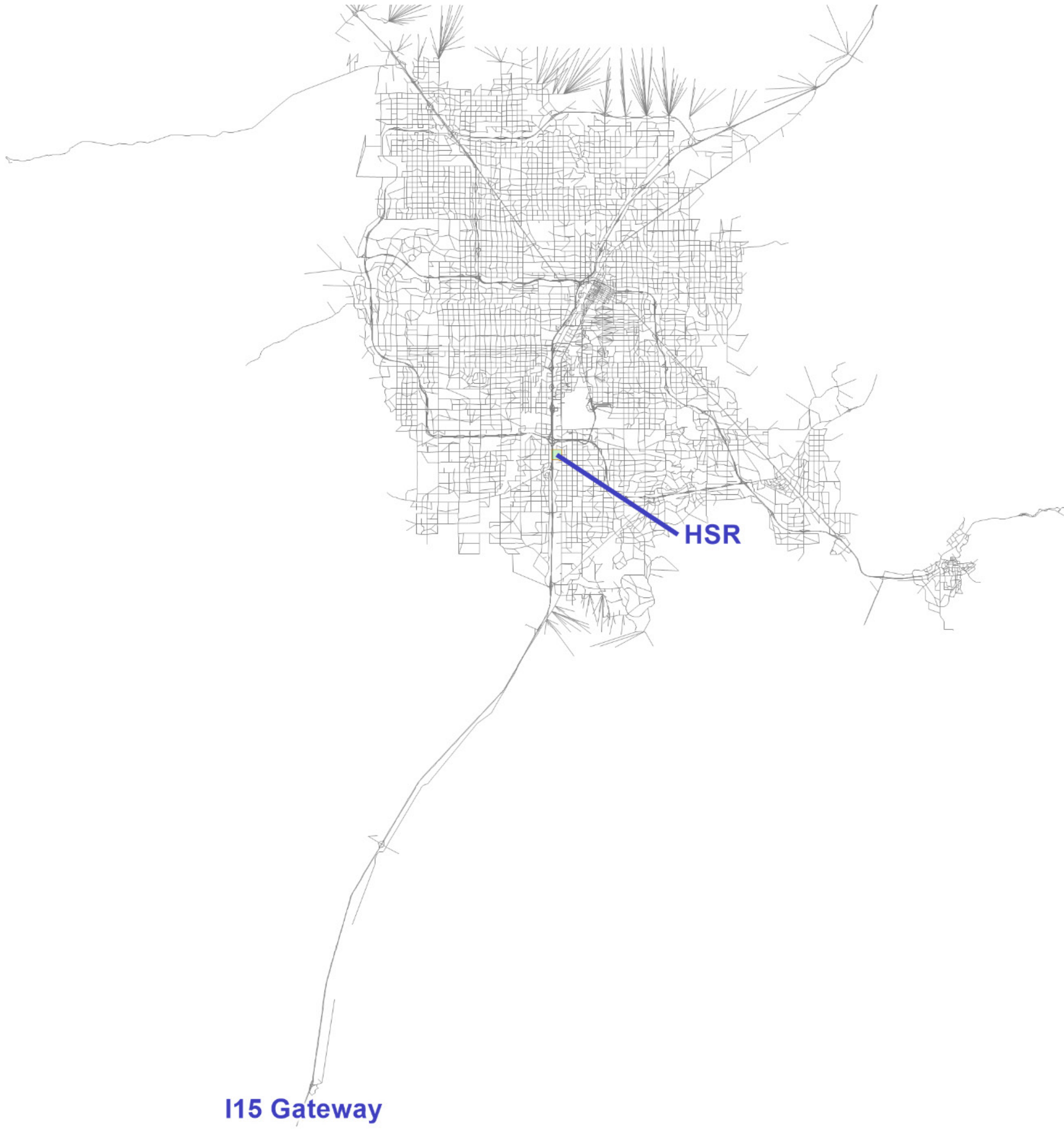


Figure 9: High-speed Rail and I15 Gateway locations

E-commerce Uncertainty

The rise of e-commerce and the consequent decline of traditional shopping trips has been a trend for many years with the rise of online shopping as an alternative to brick-and-mortar shopping. But, just as the COVID-19 pandemic highlighted unpredictable volatility in otherwise stable Visitor traffic numbers, it also underscored the role that unforeseeable world events have on phenomena such as e-commerce. Increased work from home and social distancing requirements gave way to an explosion in e-commerce well beyond the predictable trend. This uncertainty makes e-commerce a dimension ripe for EMA.

The TDM required modifications to model the effects of e-commerce in some reasonable fashion. Modeling the e-commerce trips and their impact on other decisions in the model in a trip-based framework is complicated by the need to capture the spatio-temporal nature of truck tours generated from warehouse locations in the region. Further, a more complete model implementation would need representation of the external truck trips that stock the warehouses. In other words, a more evolved model of detailed e-commerce operations would provide better sensitivity and explanatory power and hence would be better suited for EMAT. For this reason, the treatment of e-commerce in this effort is demonstrative of the kind of analysis and insight that EMAT can provide where e-commerce uncertainty is concerned, but TDM improvements are recommended before the analysis is used to inform decision-making relating to e-commerce.

E-commerce was modeled by introducing scaling factors into the TDM that reduce the number of shopping trips and replace them with truck trips. The generation of additional e-commerce truck trips takes into consideration the number of packages that replace a shopping trip, the number of packages delivered per tour, and the split between light- and medium-sized trucks. All these factors are treated as uncertainties and are included in the exploratory analysis.

Inclusion of e-commerce in the TDM consists of three main steps. First, we reduce the number of home-based shopping (HBShop) productions by an input *EcommerceDiversion* factor. Note that balancing takes care of the corresponding reduction to HBShop Attractions. Second, we replace the reduced HBShop trips by increasing the Truck Productions. We do this as follows

$$\begin{aligned}&Ecommerce\ Truck\ Productions\\&\quad = (EcommerceDiversion\ *\ [HBShop\ Productions]\\&\quad *\ PackagesPerTrip)/DeliveriesPerTour\end{aligned}$$

where *PackagesPerTrip* represents the number of packages that replace a shopping trip, and *DeliveriesPerTour* represents the number deliveries made by a single e-commerce truck tour. Note that in the TDM the truck productions and attractions are governed by the same regression equation; therefore, the same equation above also applies to e-commerce truck attractions.

Third, we add additional e-commerce truck productions (and attractions) only at the Amazon warehouse locations. This is done to model the e-commerce truck's tour start and end trips from and to the warehouse.

$$\begin{aligned}&Addtional\ Warehouse\ Ecommerce\ Productions\\&\quad = 2\\&\quad * Total\ Ecommerce\ Productions\ [Zonal\ UPS/Amazon\ warehouse\ footage]\\&\quad /[Total\ Regional\ UPS\ /\ Amazon\ warehouse\ footage]\end{aligned}$$

Based on the discussion above, uncertainty relating to e-commerce activity is captured in the EMAT application by varying the following TDM variables:

1. *EcommerceDiversion:* The percentage of home-based shopping (HBS) productions that are replaced by e-commerce.
2. *PackagesPerTrip:* The number of packages that result from dropping a HBS trip
3. *DeliveriesPerTour:* The number of package deliveries made by an e-commerce truck in a tour
4. *EcommerceTruckSplit:* The split of the e-commerce trips among light and medium trucks

Exploratory Scope and Performance Measures

In the exploratory scope, the distributions of the uncertainties described above are specified. Policy levers are also specified. However, the analysis described in this report does not explore any policy levers. The uncertainty distributions are specified in the scope file. The distributions employed are summarized in Table 2.

Table 2: Uncertainty characterization for RTC EMAT analysis with the regional TDM as core model

Uncertainty	Distribution	Minimum	Maximum
MIA_Pass	Gaussian Loc: 16700 scale: 33400	110000	233800
IVPH_Pass	uniform	8400	19600
HSR_Pass	uniform	18000	24000
EcommerceDiversion	uniform	0.1	0.3
PackagesPerTrip	uniform	4	8
DeliveriesPerTour	uniform	100	300
EcommerceTruckSplit	uniform	0.3	0.5

The minimum and maximum bounds in the visitor uncertainties were derived from extant TDM scenarios. Specifically, the 2030 scenario values were set as the minimum and 2050 scenario values were set as the maximum. The e-commerce uncertainties were approximated based on a review of news and magazine articles from media sources that have studied the rise of e-commerce.

Performance measures were chosen to capture the effects of both uncertainty dimensions. They include the Vehicle Miles Traveled (VMT), Vehicle Hours Traveled (VHT), and mode shares of the visitors. These measures were calculated for the entire region as well as for the Strip. In addition, the reduction in shopping trips, increase in truck trips, and change in truck VMT are computed to capture the effects of the e-commerce scenario. The performance measures are summarized in Table 3.

Table 3: Performance measures for RTC EMAT analysis with the regional TDM as core model

Measure name	Geography	Description
VMT_DAY_All	Entire region	Daily vehicle miles traveled
VHT_DAY_All	Entire region	Daily vehicle hours traveled
VDT_DAY_All	Entire region	Daily vehicle delay
Reduced_Shop_Trips_All	Entire region	Reduction in shopping trips
Increased_Truck_Trips_All	Entire region	Increase in truck trips
Truck_VMT_DAY_All	Entire region	Daily truck vehicle miles traveled
VIS_Total_Trips_All	Entire region	Total visitor trips
VIS_Walk_Share_All	Entire region	Visitor walk share
VIS_Taxi_Share_All	Entire region	Visitor taxi share
VIS_Public Bus_Share_All	Entire region	Visitor public bus share
VIS_Private Auto_Share_All	Entire region	Visitor private auto share
VMT_DAY_Strip	Strip	Daily vehicle miles traveled
VHT_DAY_Strip	Strip	Daily vehicle hours traveled
VDT_DAY_Strip	Strip	Daily vehicle delay
VIS_Total_Trips_Strip	Strip	Total visitor trips
VIS_Walk_Share_Strip	Strip	Visitor walk share
VIS_Taxi_Share_Strip	Strip	Visitor taxi share
VIS_Public Bus_Share_Strip	Strip	Visitor public bus share
VIS_Private Auto_Share_Strip	Strip	Visitor private auto share

Exploratory Analysis and Results

After running the experiments with the TDM as the core model, exploratory analysis was performed to understand the effects of visitor and e-commerce uncertainty on the performance measures. The measures have been normalized with respect to the default case, in which all the uncertainties are set to zero. All subsequent analysis is performed on *normalized* output measures. This will help in understanding the effect of uncertainties with results relative to the default case.

Scatter Plots

Figure 10 shows the scatter plots of different measures of VMT with respect to the uncertainty variables. From bivariate relationships we see that variables (i) MIA_Pass and EcommerceDiversion have the highest correlations with respect to daily VMT of the full region, (ii) EcommerceDiversion and DeliveriesPerTour have the highest correlations with respect to daily Truck VMT of the full region, and (iii) MIA_Pass has the highest correlation with respect to daily VMT on the Strip. This implies that increases in MIA_Pass, which represents an increase in visitors, increases VMT in the entire region, with most of the impacts felt in and around the Strip. Similarly, the truck VMT is positively affected by EcommerceDiversion, as the number of shopping trips decreases and negatively affected by DeliveriesPerTour, as higher number of delivers per tour results in lower e-commerce truck tours.

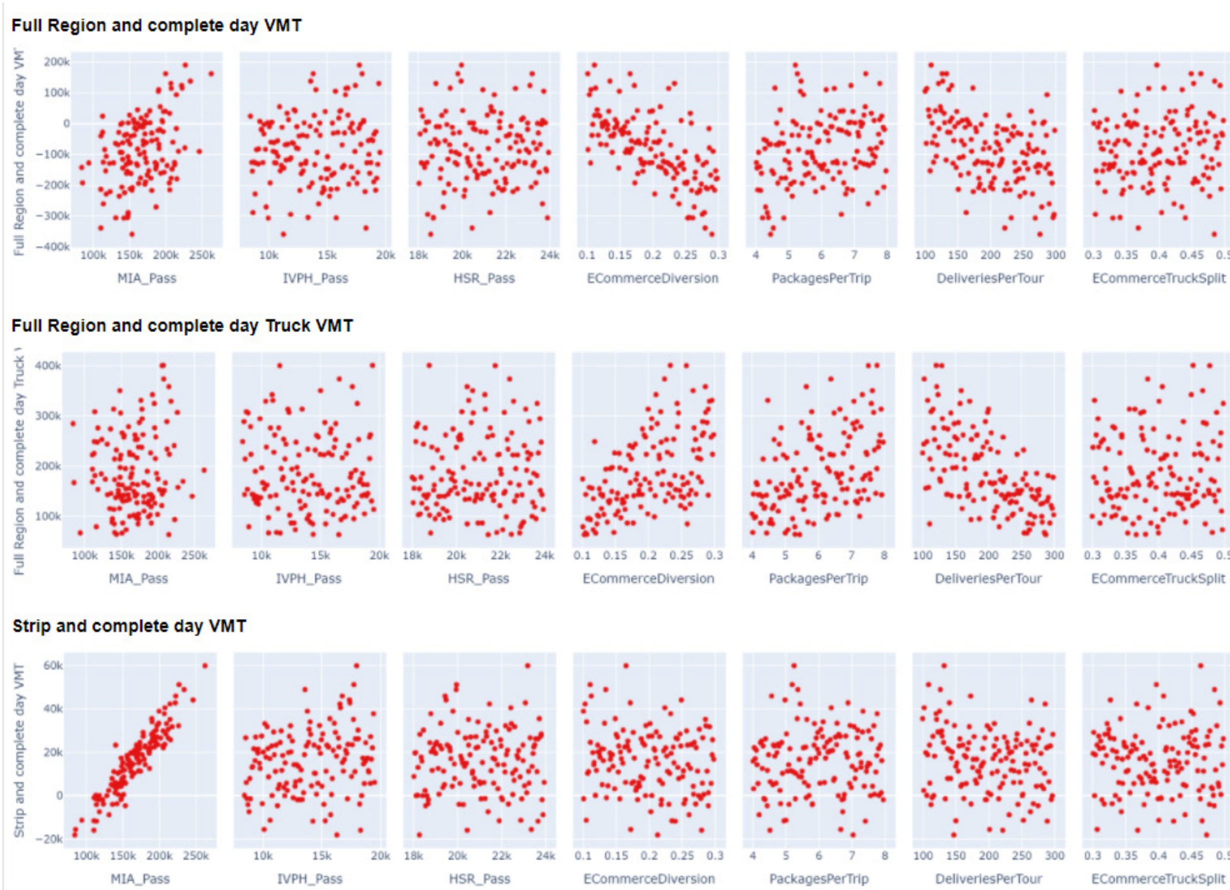


Figure 10: Scatter plots of VMT with respect to uncertainty variables

Figure 11 shows the scatter plots of changes in different trip categories with respect to the uncertainty variables. From bivariate relationships we see that (i) MIA_Pass has a strong positive (almost linear) relationship with respect to the total visitor trips, (ii) EcommerceDiversion has a strong linear relationship with respect to the reduction in shopping trips, and (iii) EcommerceDiversion and DeliveriesPerTour have the highest correlations with respect to daily truck trips.

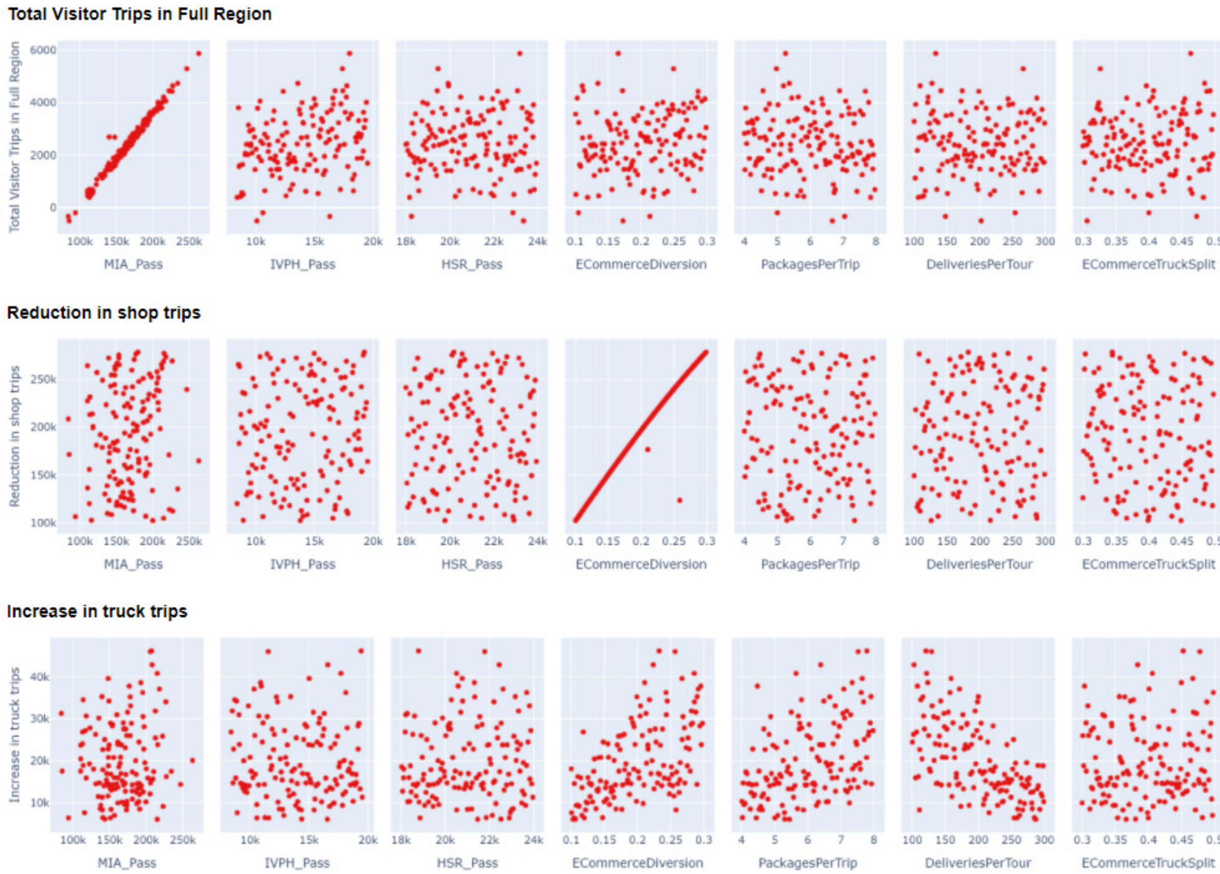


Figure 11: Scatter plots of trips with respect to uncertainty variables

As MIA_Pass is directly related to the number of visitor trips, the strong positive relationship with the total number of visitor trips is expected. EcommerceDiversion is directly related to thereduction in shop trips by scenario design which was discussed above. Increase in EcommerceDiversion increases truck trips because of E-commerce demand and increase in DeliveriesPerTour decreases truck trips as lower number of E-commerce truck trips are required to meet the demand.

Figure 12: Scatter plots of visitor mode shares (in %) with respect to uncertainty variables.

Figure 12 shows the scatter plots of changes in visitor mode shares with respect to the uncertainty variables. From the y-axis ticks, we see that mode shares vary very little with respect to the uncertainty variables. This is because the uncertainties above (and their ranges) only marginally affect the utilities of visitor mode alternatives, which, in turn, is likely due to the structure of the visitor mode choice model itself (e.g., the multiplicity of coefficients and parameters involved).

Feature scores

Figure 13 is a visualization of the features scored for all output measures. For a given output measure, the important (i.e., most influential) input uncertainty variables are highlighted. For example, the daily VMT, VHT and VDT are most influenced by EcommerceDiversion, DeliveriesPerTour, and MIA_Pass. Similarly, daily truck VMT is influenced most by DeliveriesPerTour followed by EcommerceDiversion; it depends less on PackagesPerTrip.

	DeliveriesPerTour	ECommerceDiversion	ECommerceTruckSplit	HSR_Pass	IVPH_Pass	MIA_Pass	PackagesPerTrip
VMT_DAY_All	0.192747	0.333833	0.065039	0.068238	0.072731	0.160680	0.106732
VHT_DAY_All	0.176686	0.331917	0.078270	0.062522	0.067650	0.184644	0.098311
VDT_DAY_All	0.225291	0.173295	0.111820	0.064059	0.077088	0.248276	0.100172
VIS_Total_Trips_All	0.068252	0.067177	0.069749	0.064270	0.109698	0.557634	0.063220
VIS_Walk_Share_All	0.062288	0.066624	0.062424	0.068043	0.097387	0.575760	0.067474
VIS_Taxi_Share_All	0.053973	0.057422	0.059167	0.057822	0.539416	0.174840	0.057359
VIS_Public Bus_Share_All	0.057286	0.059861	0.057589	0.052932	0.432412	0.290461	0.049460
VIS_Private Auto_Share_All	0.063955	0.065704	0.068223	0.083321	0.388945	0.261093	0.068757
Reduced_Shop_Trips_All	0.061602	0.660705	0.053865	0.054167	0.056658	0.062607	0.050395
Increased_Truck_Trips_All	0.283466	0.263066	0.077239	0.062608	0.068491	0.081007	0.164121
Truck_VMT_DAY_All	0.305297	0.254019	0.066426	0.063991	0.062900	0.073961	0.173407
VMT_DAY_Strip	0.083988	0.094718	0.066514	0.077660	0.090354	0.517833	0.068934
VHT_DAY_Strip	0.100723	0.128983	0.077671	0.082991	0.093881	0.445431	0.070319
VDT_DAY_Strip	0.097334	0.152620	0.084560	0.072160	0.079284	0.442560	0.071482
VIS_Total_Trips_Strip	0.068558	0.068256	0.064974	0.072889	0.107922	0.554067	0.063334
VIS_Walk_Share_Strip	0.064304	0.058675	0.060078	0.069844	0.098067	0.589321	0.059712
VIS_Taxi_Share_Strip	0.061335	0.068309	0.060961	0.071384	0.089827	0.587699	0.060485
VIS_Public Bus_Share_Strip	0.060353	0.073641	0.062074	0.064699	0.097703	0.578519	0.063011
VIS_Private Auto_Share_Strip	0.056623	0.063443	0.059123	0.065292	0.094690	0.606758	0.054071

Figure 13: Features scores for uncertainty variables in the regional TDM

The feature scores also indicate that visitor mode shares are affected mostly by MIA_Pass and IVPH_Pass, which was also observed in the scatter plots previously described. The output measures for the Strip, including VMT, VHT, VDT and visitor mode shares, are influenced predominantly by MIA_Pass.

Figure 14 is a violin plot of the threshold feature scores for daily VMT. The plot shows that the increase in VMT, with respect to the default case, is explained mostly by MIA_Pass. In other words, MIA_Pass contributes the most to an increase in VMT from the default case. Similarly, we see that EcommerceDiversion contributes most to the decrease in VMT with respect to the default case. Figure 14 also shows some marginal effect from the DeliveriesPerTour and PackagesPerTrip variables.

Figure 15 is a violin plot of the threshold feature scores for daily truck VMT. The plot shows that truck VMT always increases with respect to the default case, which is expected because the default value of EcommerceDiversion is zero. The most influential variables explaining the increase in truck VMT with respect to the default case are EcommerceDiversion, DeliveriesPerTour, and PacakgesPerTrip.

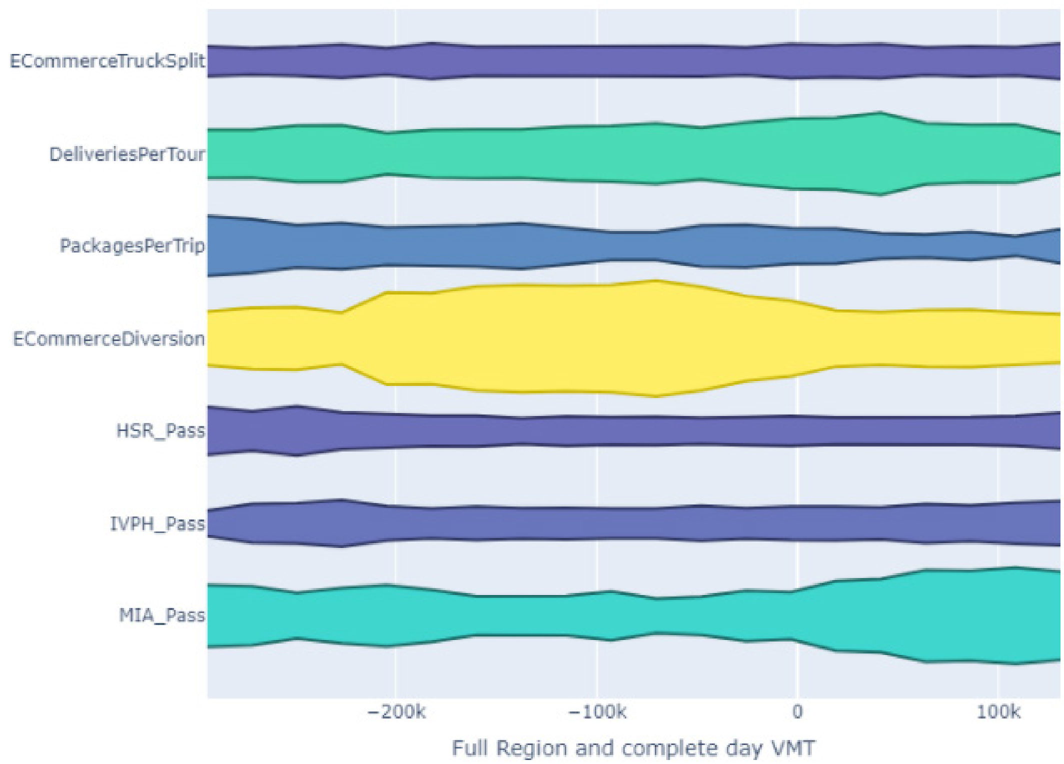


Figure 14: Threshold Feature Scores for Daily VMT

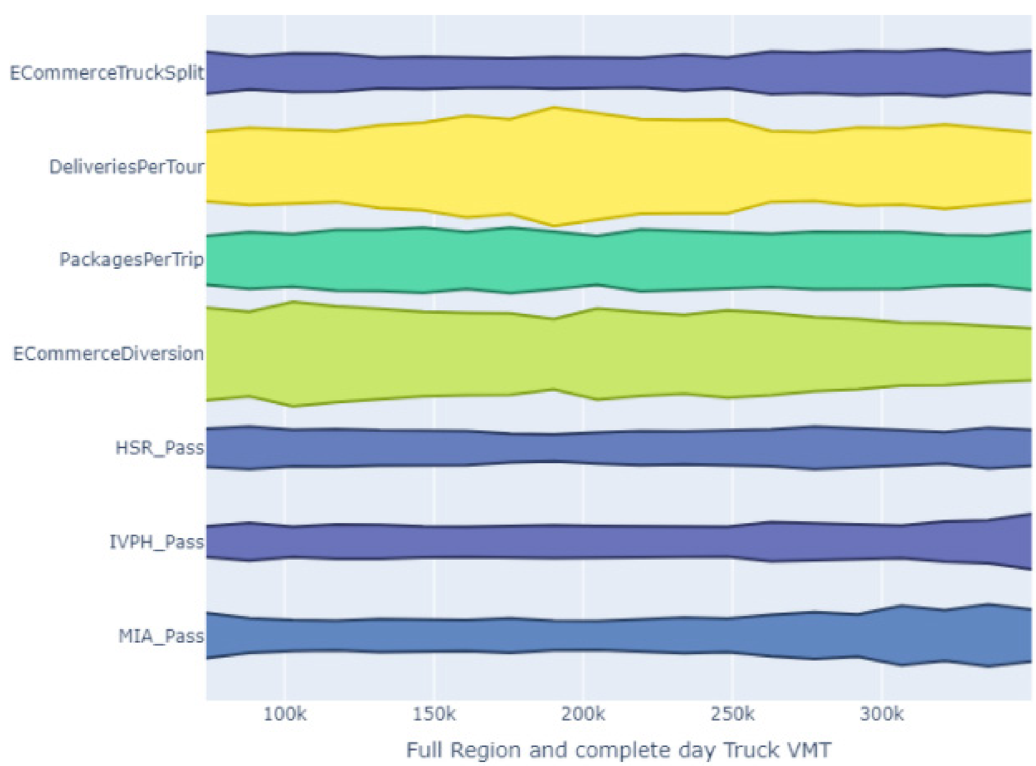


Figure 15: Threshold Feature Scores for Daily Truck VMT

5.0 EXPLORATORY ANALYSIS WITH MICROSIMULATION-BASED DTA

The EMAT application integrated with TransCAD and TransModeler was used to explore various operational and demand uncertainties and mitigation measures relating to increased traffic destined for T-Mobile Arena, which is located at the I-15-Tropicana Ave interchange between Las Vegas Blvd to the east and Frank Sinatra Dr to the west (Figure 16).

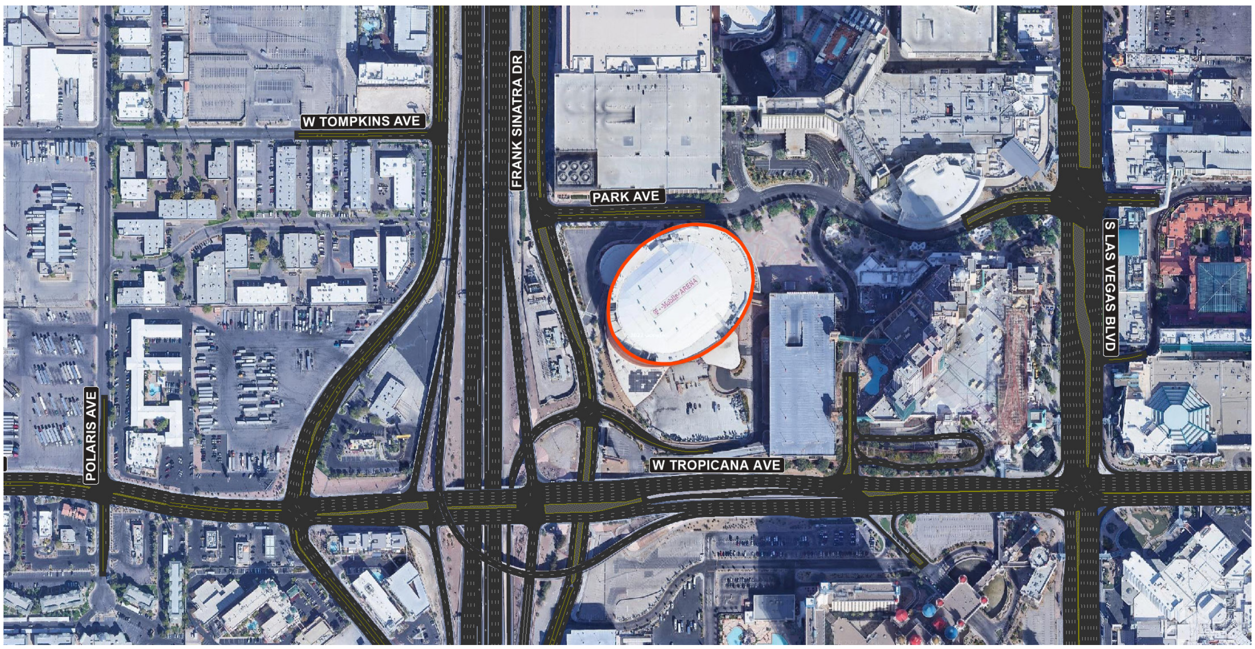


Figure 16. Map of T-Mobile Arena and surrounding area

Specifically, the analysis considers a scenario in which a Las Vegas Golden Knights (VGK) hockey game is scheduled to begin at 6:00 PM on a weekday. 6:00 PM coincides with the end of the PM peak period.

Two EMAT applications with DTA were tested. The first used a regional microsimulation-based DTA model previously developed and calibrated for the RTC in TransModeler as the starting point for the core model. In the model, performing an assignment for a single experiment calls for a microsimulation of the entire region, which spans Clark County's borders and beyond. The model of the PM peak period simulates almost 1.9 million trips in a 5-hour period from 1:00 to 6:00 PM. On a capable desktop PC, each iteration of an assignment will take approximately 20 minutes to complete, and convergence will typically be achieved in 50 or fewer iterations, depending on the routing inputs provided to the DTA. Hence, one assignment will take about 17 hours to run.

Seven "factors" – uncertainty variables and policy levers – were included in the analysis. Choosing a sample size of 10 per factor (n_samples_per_factor in the TMIP-EMAT API), the default recommended in TMIP-EMAT guidance, required that 70 assignments be performed. 70 assignments would require about days hours to complete unless they can be distributed among different computers, an extension to the application implementation that was not attempted as part of this work.

To achieve more tolerable running times, a subarea around T-Mobile Arena was defined that would capture all of the alternative routes to I-15 and Tropicana Ave, from Decatur Blvd in the east to Maryland Pkwy in the east and from Bonanza Rd (north of US 95) in the north to Warm Springs Rd (south of I-215). The subarea, depicted in Figure 17, is about four miles from east to west and 10 miles from north to south, or 40 square miles. The simulation period was also reduced to a period of time closer to the start of the game, beginning at 3:00 PM and continuing to 6:00 PM. About 485,000 trips are simulated in the 3-hour period. The simulation is started with traffic preloaded in the network so that no warm-up period is required.

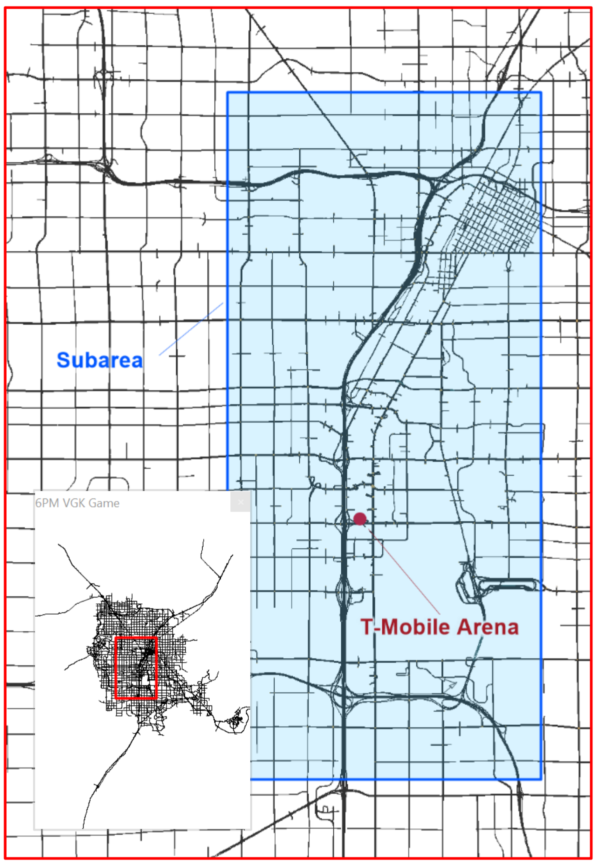


Figure 17. Boundaries of the microsimulation-based DTA subarea

The reduction in study area size and simulation period resulted in running times of about 8 minutes per DTA iteration. The reduced study area size and the use of warm-start congested travel times and delays as inputs to the route choice model together allowed for fewer DTA iterations to be run. 15 iterations were run in each experiment in order to converge to stable route choices in which simulated times are consistent with expected travel times. As a result, each DTA experiment took about 2 hours to run, and the entirety of 70 EMAT experiments was completed in about 6 days' total running time.

Uncertainty Variables

The primary purpose of the application is to demonstrate EMAT not only as a tool for planners faced with the uncertainties inherent in travel demand modeling but as a tool to help planners and engineers answer difficult questions relating to traffic operations and maintenance of traffic. The uncertainty variables chosen for the analysis are summarized below:

1. Background Traffic: A coefficient of variation (CV) used to reflect the inherent randomness of background traffic (i.e., those trips not destined for T-Mobile Arena) from day to day
2. Attendance: The number of auto trips attracted to T-Mobile Arena for the VGK game
3. Dynamic Message Sign (DMS) Compliance: Percentage of drivers *not* attending the game that heed a DMS message on I-15 encouraging use of alternative interchanges to Tropicana
4. CAV Percentage: Percentage of the vehicle fleet that are connected/autonomous vehicles (CAV)

Background Traffic Uncertainty

The microsimulation of the PM peak in the original DTA model is calibrated to represent recurring weekday traffic patterns according to historical traffic count and speed data throughout the region. However, traffic volumes are measurably varied from day to day. Because modest variation in background traffic may impact operations near the arena and the efficacy of mitigation measures, the variance in day-to-day traffic was included as a variable of uncertainty. In TransModeler, the number of trips that are simulated between any origin-destination (O-D) pair is provided as an input in a time-varying O-D matrix. These volumes are assumed to represent the average numbers of trips traveling between the O-D pair on any given day. Each simulation can be considered to represent one realization of the O-D pattern, or one day in the real world. In TransModeler, a CV can be specified by the user to approximate the variance in real trips across days. The CV is the ratio of the standard deviation in number of trips to the mean number of trips and is used to randomize the number of trips simulated in any given simulation.

Attendance Uncertainty

While data were available to estimate the VGK game attendance, which numbers around 18,000, very little data were available to estimate the number of driving trips attracted by a game. Part of the challenge is the multitude of parking options available to fans attending the game. These options include garages at the MGM Grand and New York-New York casinos, which neighbor T-Mobile Arena. Numerous factors were considered when deciding the limits on the likely number of auto trips to the arena, including:

- a significant number of trips are likely to have two or more fans attending the game as a couple or as a group
- a significant number attending may be visitors to other destinations on the strip and would not drive to the arena

The trips traveling to T-Mobile Arena were modeled as trips traveling to the traffic analysis zone (TAZ) in which the arena is located and were assumed to have origins distributed throughout the region in the same proportion as background traffic to the TAZ. Game trips were treated as new trips occurring during the PM peak, departing as early as 1.5 hours before puck drop. Hence, a game trip matrix was created whose trips are distributed between 4:30-6:00 PM and whose numbers could be scaled proportionally such that the total number of trips generated from the game matrix matched the value drawn for the EMAT experiment.

DMS Compliance Uncertainty

Many DMSs were newly installed on I-15 as part of Project Neon. These DMSs are part of FAST's ITS infrastructure and a potential resource for mitigating the effects of increased traffic at the Tropicana interchange due to a VGK game. In this project, the assumption is made that these DMSs may be enlisted to inform drivers on I-15 of increased traffic and delays at the Tropicana Ave interchange.

In the northbound direction, two DMSs are located on I-15 just south of the Sunset Rd overpass, approximately two miles south of Tropicana Ave and about ¾ mile before drivers would exit I-15 for Tropicana Ave. One DMS is installed over the I-15 mainline and the second over the parallel frontage lanes. In the southbound direction, the nearest DMS upstream of Tropicana Ave is a solitary DMS located south of the Flamingo Rd overpass between the exit and entrance ramps for Flamingo Rd. The southbound DMS is about 0.8 miles from Tropicana Ave and about ½ mile from the exit ramp to Tropicana Ave.

In TransModeler, three variable message signs (VMS), one for each physical DMS, were placed at the corresponding locations. The locations of the northbound DMSs in the DTA model are depicted in Figure 18.

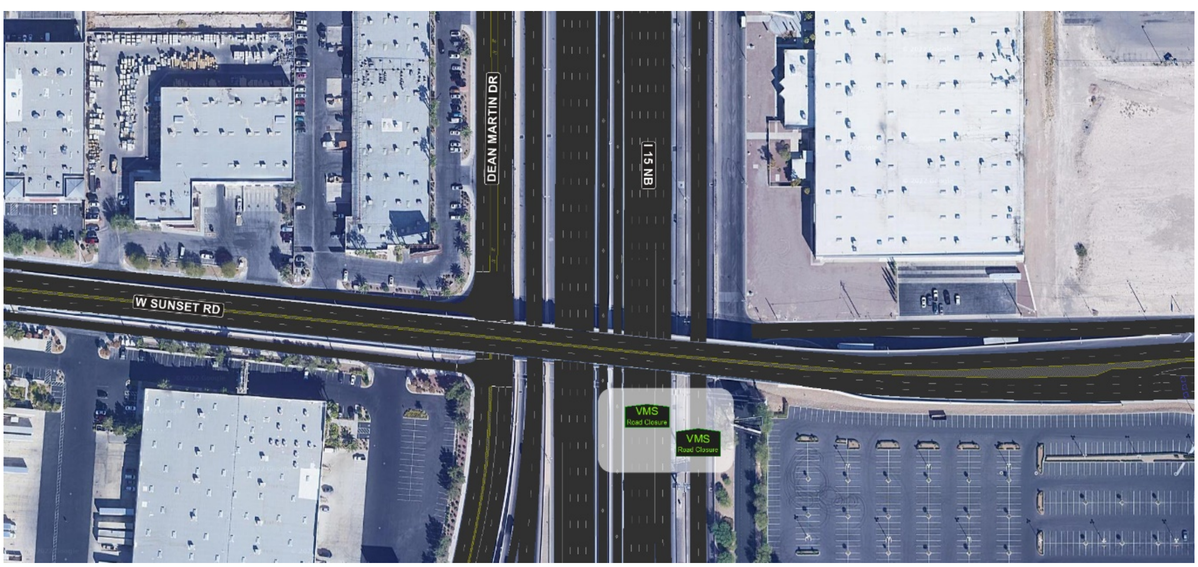


Figure 18. Northbound variable message signs (VMS) on I-15 in the DTA model

The VMSs used in the microsimulation-based DTA in TransModeler were road closure messages, which need not necessarily convey that a road is closed to all traffic. Rather, a road closure VMS may be used to close roads to a specified fraction of vehicles or to a specific class of vehicle or trip. The analyst may posit the percentage of drivers that comply with the message and enter this percentage as an input compliance rate. In other words, the VMS is used to identify (1) the target roadway links and (2) the percentage of drivers that should seek alternative routes to avoid those links. The remaining drivers will ignore the message and consider the roads to be open. The target link in the southbound direction is the Tropicana Ave exit ramp. In the northbound direction, the target link is the frontage road just south of the left-side diverge to Arena Dr.

The percentage of drivers that comply with the DMS and seek alternative routes to Tropicana Ave is an unknown and is treated in the EMAT experiments as a variable of uncertainty. Further, the VMSs in the model were made to target only drivers not attending the game at T-Mobile Arena. All trips to the arena for the game would ignore the VMSs and hence be eligible to use the Tropicana Ave interchange.

CAV Percentage Uncertainty

Though CAVs are not specifically a mitigation strategy for managing VGK game traffic at T-Mobile Arena, CAVs may have general and incidental benefits with respect to increased traffic due to events such as VGK games. To explore what those benefits might be if some percentage of vehicles on the roadway today were CAVs capable of forming platoons with short following headways, CAV percentage was also introduced as a variable of uncertainty.

Policy Levers

To highlight the operational advantages of DTA and the potential advantages of integrating operational modeling fidelity with EMA, several policy levers were chosen that targeted traffic operations and management. While input was sought from FAST, which holds the responsibility for managing game and event traffic in and around T-Mobile Arena, the strategies largely centered on police officers directing traffic at key signalized intersections near the arena. The simulation model in the DTA does not have a model for emulating police officer behavior and decision-making, so this strategy was not an option in the EMAT experiments. Instead, the following approaches were assumed to be reasonable policy actions within FAST's means and resources even if they are not presently a part of its event management or might face implementation challenges not considered as part of this demonstration exercise:

1. Road Closure
2. Ramp Metering at Tropicana
3. Advisory DMS Messaging

Road Closure

Road closures near T-Mobile Arena, though not necessarily an impact mitigation strategy, may have other benefits, such as improving pedestrian safety and encouraging alternative modes of travel to the Arena, and will have operational impacts. Four road closure policies were considered:

1. Do Nothing: No roads are closed

2. Park: Segments of Park Ave between Frank Sinatra Dr in the west and Las Vegas Blvd to the east are closed to all vehicle traffic
3. Frank Sinatra: The segment of Frank Sinatra Dr between Park Ave in the north and Arena Dr to the south is closed to all vehicle traffic
4. Park and Frank Sinatra: All segments of Park Ave and Frank Sinatra Dr described above are closed to all vehicle traffic

Ramp Metering at Tropicana

Ramp meters are potentially a tool to address traffic congestion concerns on Tropicana Ave when there is increased traffic at the interchange due to a VGK game. By shutting the ramp meters on the ramps from Tropicana Ave to I-15 northbound and southbound, queues that may spill back from those meters to Tropicana Ave can be avoided and increased traffic may be more easily cleared from the area. However, shutting off the ramp meters may also introduce or exacerbate bottlenecks on I-15 at the merge from the Tropicana Ave entrance ramps. Hence, to ensure that the mitigation steps do not benefit traffic flow on Tropicana Ave at the expense of flow on the interstate, performance measures were chosen that distinguish between freeway and arterial impacts. More about the specific performance measures will follow.

The ramp metering policy lever took on one of two categorical values in the EMAT experiments: ON or OFF. When ramp metering was on, it followed the normal schedule and operating parameters. When ramp metering is off, the ramp meters are shut off at 5:00 PM, one hour before both the start time of the game and the normally scheduled shut off time.

Advisory DMS Messaging

DMSs on I-15 are another resource that could potentially be leveraged to mitigate the impacts of VGK game traffic. As described previously in this report. DMSs were placed in the DTA model where they exist in the field. These DMSs could be used to warn travelers on I-15 of increased delays due to the VGK game. A message could display the game start time and alert drivers to expect delays. Such a sign may encourage some proportion of drivers to consider alternative routes if another exit could be used to reach their destination. Road closure DMSs in TransModeler prompt drivers, all or a specified proportion of them, to seek new routes that avoid a specified road or multiple roads.

The DMS policy lever took on one of two categorical values in the EMAT experiments: YES or NO. When DMSs are in use (i.e., value YES), they display a message prompting some percentage of travelers who are not attending the game to take a route avoiding the Tropicana Ave exit, where the percentage is an uncertainty variable. When DMSs are not in use (i.e., value NO), the DMSs display no message and have no influence on driver behavior.

Exploratory Scope and Performance Measures

In the exploratory scope, the distributions of the uncertainties described above are specified. Policy levers are also specified. The uncertainty and policy lever distributions, or categorical values, are specified in the scope file. The uncertainty distributions employed are summarized in Table 4, and the policy levers and categorical values are listed in Table 5.

Table 4: Uncertainty characterization for RTC EMAT analysis with the microscopic DTA as core model

Factor	Distribution	Minimum	Maximum
BackgroundTraffic	uniform	0%	5%
Attendance	uniform	2600	6000
DMSCompliance	uniform	0%	50%
CAV	uniform	0%	100%

Table 5. Policy levers for the RTC EMAT analysis with the microscopic DTA as core model

Factor	Values
Closure	Do Nothing / Park / Frank Sinatra / Park and Frank Sinatra
RampMetering	ON / OFF
DMS	NO / YES

Little is known empirically about the minimum and maximum values for the uncertainty variables, with the exception of game attendance. As discussed previously, attendance numbers from past VGK games are known, but the numbers of attendees who driver to the game and park at the MGM Grand and New York-New York casinos or other immediately adjacent parking garages is not known, though data to help estimate the number of vehicle trips were sought. Reasonable limits on the other uncertainty variables were asserted.

Performance measures were chosen that would help capture the overall effects of the uncertainty variables and policy measures and that would help distinguish between impacts on freeway and arterial operations and between impacts felt by trips traveling to T-Mobile Arena to attend the game and background trips. A distinction between freeway and arterial performance and between game and background trips is needed because certain policy measures will impact the performance of some facilities more directly than others or will have impacts that are felt more directly in the experiences of some travelers than others.

For example, ramp metering may directly benefit operations on Tropicana Ave, a principal arterial, but may negatively influence operations on I-15. Similarly, DMSs may benefit travelers to T-Mobile Arena but may negatively impact the speeds and delays experienced by background traffic as travelers that would normally use the Tropicana Ave interchange divert to other interchanges, potentially overburdening adjacent routes and facilities.

To capture all of these performance measures, scripts had to be written in the TransModeler API (i.e., GISDK) to pull specific metrics from the raw simulation outputs that are meaningful in the context of the VGK game scenario. In other words, the performance measures are not measures TransModeler routinely produces but are custom and rely on analyst familiarity with the API tools in TransModeler.

One of these measures is travel rate, a measure that has seen increased interest in the context of travel reliability analysis. The travel rate, often expressed in minutes per mile, is interesting in the reliability context for the same reason it was chosen as an EMAT performance measure: it is a distance-neutral surrogate for trip travel time variance. Different trips impacted by the uncertainty variables and policy levers have widely varying lengths, making it difficult to identify measures that can be averaged across them to meaningfully summarize their collective experiences. The travel rate is, effectively, the average time it takes to travel a unit distance. An increase in the magnitude of the travel rate corresponds with a decline in level of service. In other words, low travel rates are desirable.

Table 6: Performance measures for RTC EMAT analysis with the microscopic DTA as core model

Measure name	Geography	Trips	Description
VMT	Entire model subarea	All trips in subarea	Vehicle miles traveled
VHT	Entire model subarea	All trips in subarea	Vehicle hours traveled
CompletedTrips	Entire model subarea	All trips in subarea	Total number of trips successfully completed
FreewayDelay	On the freeway system (primarily I-15), including ramps	All trips	Vehicle hours of delay experienced on freeways and ramps
NonFreewayDelay	On surface streets/ All trips	All trips	Vehicle hours of delay experienced on surface streets
TravelRateBackground	Entire model subarea	Background trips	Minutes per mile traveled
TravelRateGame	Entire model subarea	Game trips to T-Mobile Arena	Minutes per mile traveled

All performance measures are summarized from the entire period of simulation between 3:00 and 6:00 PM.

VMT and VHT do not directly describe level or quality of service but are easily obtained from the simulation outputs and can be useful in discerning overall trends. Similarly, the number of completed trips is not directly a measure of performance but can point to increased congestion. For example, if fewer trips in one experiment are able to be successfully completed relative to other experiments, it may be deduced that trips experienced increased delays and hence were still en route when the simulation finished.

Exploratory Analysis and Results

After running the experiments with the microscopic DTA in TransModeler as the core model, exploratory analysis was performed to understand the effects of the factors - uncertainty variables and policy levers – scoped for the analysis on the performance measures also identified in the experimental design.

The experiments were run multiple times. Each time, adjustments were made based on study of the performance measures of the prior experiments. On some occasions, those adjustments were made to correct errors in the model inputs. On other occasions, adjustments were made to improve running times.

In that process, one conclusion was quickly drawn: the CV of background traffic had little to no influence over any of the performance measures. In hindsight, this is a reasonable outcome and led to an important lesson learned: not all variables about which we are uncertain are necessarily good candidates for uncertainty variables in EMA.

The CV of background traffic, whether high or low, impacts the numbers of trips in both directions: some origin-destination pairs will produce more trips than the average day and some will produce fewer. In the aggregate, increases in volumes and decreases in volumes may balance one another out, leading to negligible net effects. Moreover, because there is no spatial or temporal pattern in the demand increases and decreases, only highly localized or brief aberrations in performance are likely to result. Hence, there is no a priori reason to expect a high or low CV of background traffic demand to push performance measures in one direction or the other. In other words, responses to the CV are likely to be highly varied and operationally neutral.

In the later iterations of the experiments that were run, *BackgroundTraffic* was removed from the experimental design and hence from the scope file. From these results, there are two ways for an analyst to quickly glean insights from the considerable volume of data produced by the experiments. First, a heat map summarizing feature scores (Figure 19) helps the analyst discern the most impactful factors from among the uncertainty variables and policy levers explored. Second, an array of bivariate scatter plots displayed in the EMAT Dashboard allows the analyst to quickly scan model outcomes for relationships between all combinations of factor and performance measure.

	ATTENDANCE	CAV	CLOSURE	DMS	DMSCOMPLIANCE	RAMPMETERING
CompletedTrips	0.094	0.567	0.138	0.049	0.107	0.045
FreewayDelay	0.199	0.256	0.218	0.079	0.182	0.067
NonFreewayDelay	0.201	0.165	0.366	0.054	0.144	0.07
TravelRateBackground	0.157	0.33	0.22	0.054	0.17	0.069
TravelRateGame	0.157	0.172	0.393	0.069	0.141	0.067
VHT	0.24	0.232	0.195	0.076	0.198	0.059
VMT	0.113	0.563	0.125	0.05	0.104	0.045

Figure 19. Features scores for uncertainty variables in the microscopic DTA

Put another way, Figure 19 helps the analyst to identify where the explanatory power in the model outcomes lies, and Figure 20 helps the analyst understand the direction of the influence between individual factors and individual performance measures. These visualizations can serve as points of departure for deeper explorations of the relationships between factors and performance measures, which may reveal simple correlations or more nuanced compound or cancelation effects of multiple factors on model outcomes.

Or, if these results do not confirm a priori expectations, then scrutiny of the modeling assumptions, model sensitivities, and scope configuration parameters may be warranted before further analysis of the results is undertaken.

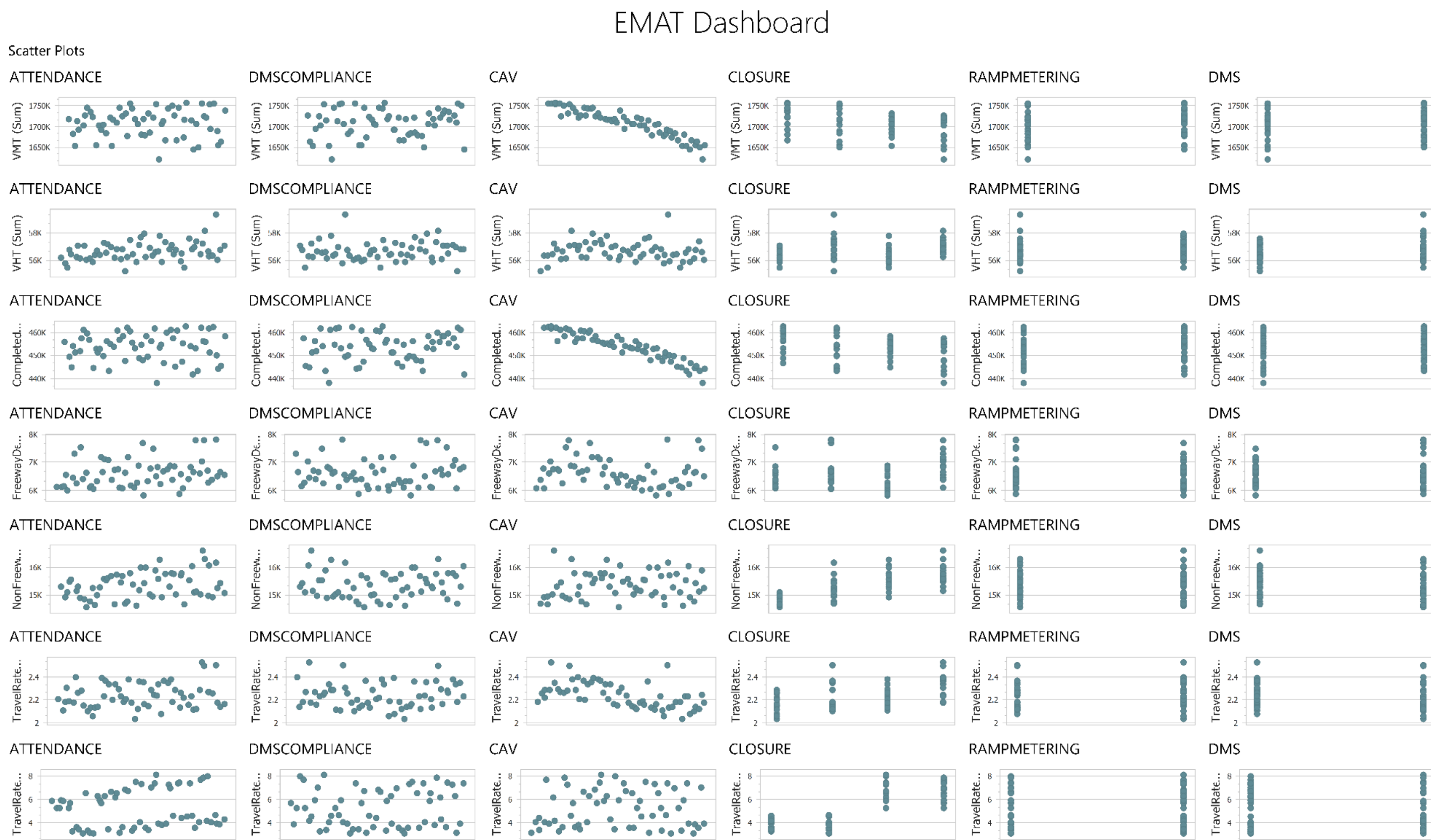


Figure 20. Scatter plots between experimental factors and performance measures in the EMAT Dashboard

Upon reviewing the results of the EMAT experiments with the microscopic DTA model, the feature scores draw attention first to the size of the influence of CAV percentage on two performance measures: *CompletedTrips* and *VMT*. To a lesser extent, some influence can be seen on *TravelRateBackground* and *FreewayDelay*. A logical next step is to consider the scatter plots between these four performance measures and CAV percentage. To that end, Figure 20 reveals clear trends in declining VMT and completed trips as the CAV percentage increases and suggests potential trends in freeway delay (i.e., *FreewayDelay*) and the travel rates of background trips (i.e., *TravelRateBackground*). A closer inspection of the latter scatter plots from Figure 20 is provided in Figure 21 and Figure 22.

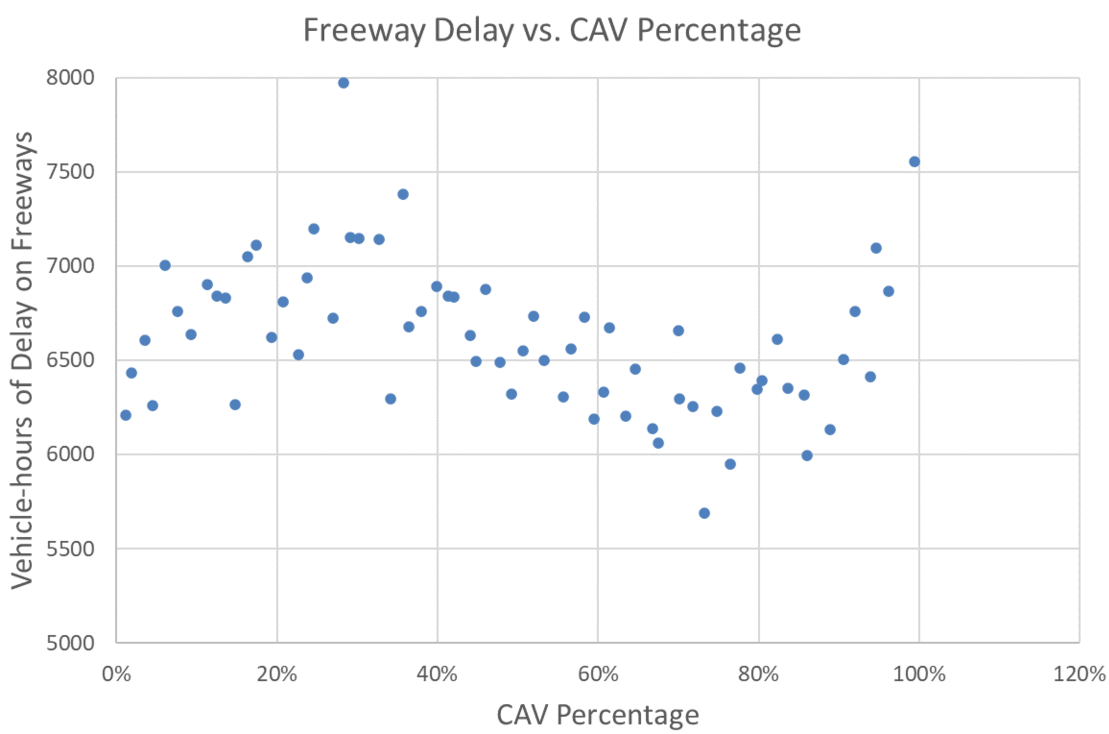


Figure 21. Scatter plot of freeway delay versus CAV percentage

Figure 21 and Figure 22 do not show as strong a correlation between freeway delay and the travel rate of background trips, but there is a general decline in those performance measures as the CAV percentage increases that would warrant deeper analysis if the effects of CAVs and their interplay with other factors were of serious interest to the RTC.

The increase in VMT with increasing CAV percentage is curious and worthy of further consideration. With the exception of the game demand, the background demand is constant across all experiments. The fact that VHT remains steady indicates that trips are spending the same amount of time traveling shorter distances. Fewer vehicles are also successfully completing their trips as CAV percentage increases, which is consistent with the observed decline in VMT. These observations would suggest that trips are experiencing more delay, spending greater lengths of time in queues and/or in slow-moving traffic. However, the trends in freeway delay and background travel rate suggest otherwise, the latter suggesting that trips that are not traveling to the game experience less travel time per mile traveled, though this decline is modest.

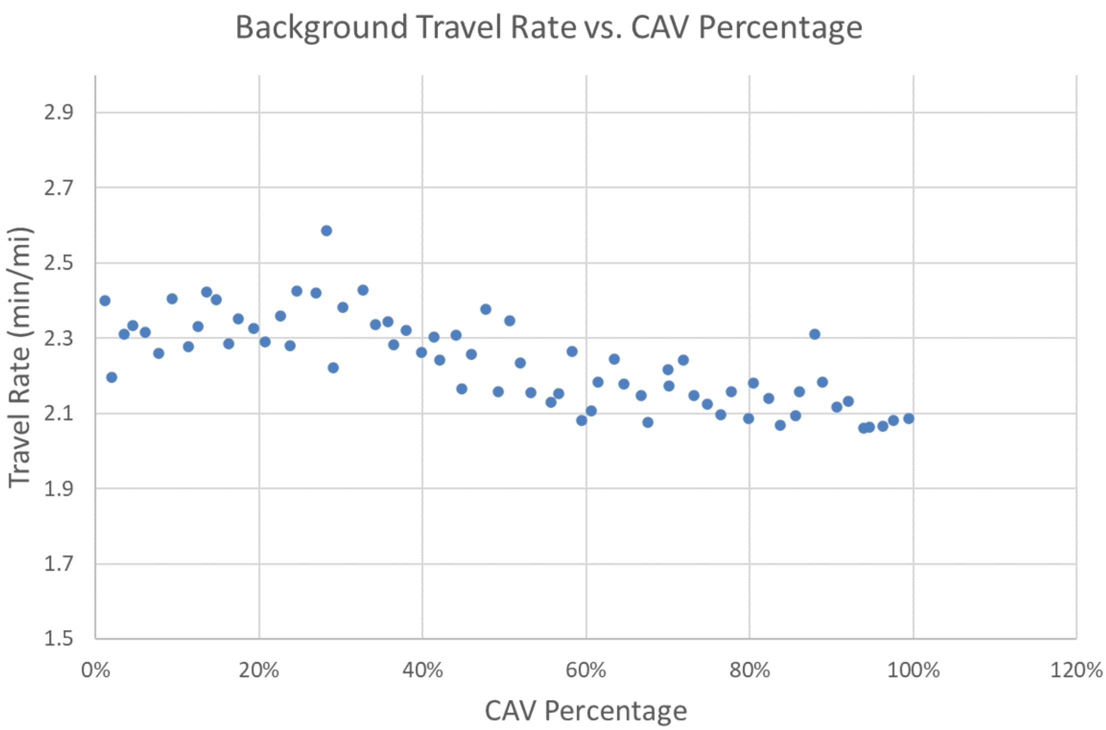


Figure 22. Scatter plot of background travel rate versus CAV percentage

These seemingly contradictory findings suggest the influence of a confounding variable not part of the analysis scope. To identify this variable, a visual inspection of a simulation with a high percentage of CAVs was performed. This inspection quickly revealed an experimental design not fully accommodating of CAVs. In the experiments, it was assumed that all CAVs were capable of participating in platoons of up to 12 vehicles in any lane anywhere in the network. However, these platoons may have acted as moving barriers making it exceedingly difficult for vehicles to execute their weaving or turning maneuvers. After missing a turn along a vehicle's path, it may be forced to leave the network early if an alternate path does not exist to the destination. These trips that fail to traverse their entire paths are not counted among the number successfully completed. Further, these trips would not go on to contribute to higher volumes on freeway facilities downstream. This would have the effect of reducing traffic volumes elsewhere in the network, leading to lower freeway delays and travel rates for background trips.

In other words, the lack of an effective CAV management policy, such as limiting platoon lengths or restricting on which facilities or in which lanes vehicles may form platoons, in the experimental design led to a high incidence of failed trips, fewer overall miles traveled, and, as a consequence, reduced delays.

Apart from the CAV factor, the road closure policy lever has the most influence, with impacts felt most on surface streets and by background traffic that must seek alternative routes as a consequence of the road closure policy. These outcomes are unsurprising, given that road closures reduce capacity. Streets that remain open and that serve as alternative routes will bear the additional traffic load as drivers shift their routes. This can be seen in Figure 23, which depicts the increasing travel rate of background traffic with various closure policies, Figure 24, which depicts the increasing non-freeway (i.e., surface street) delay with different closure policies, and, lastly, Figure 25, which illustrates how non-freeway delay increases as game attendance grows.

These outcomes comport with the following expectations:

1. Delay increases as the number of vehicle trips attending the game increases and hence the traffic demand on surface streets feeding parking facilities near the arena increases.
2. All traffic suffers higher travel rates and surface streets see increased delays as road closures near the arena push other streets closer to or beyond saturation.

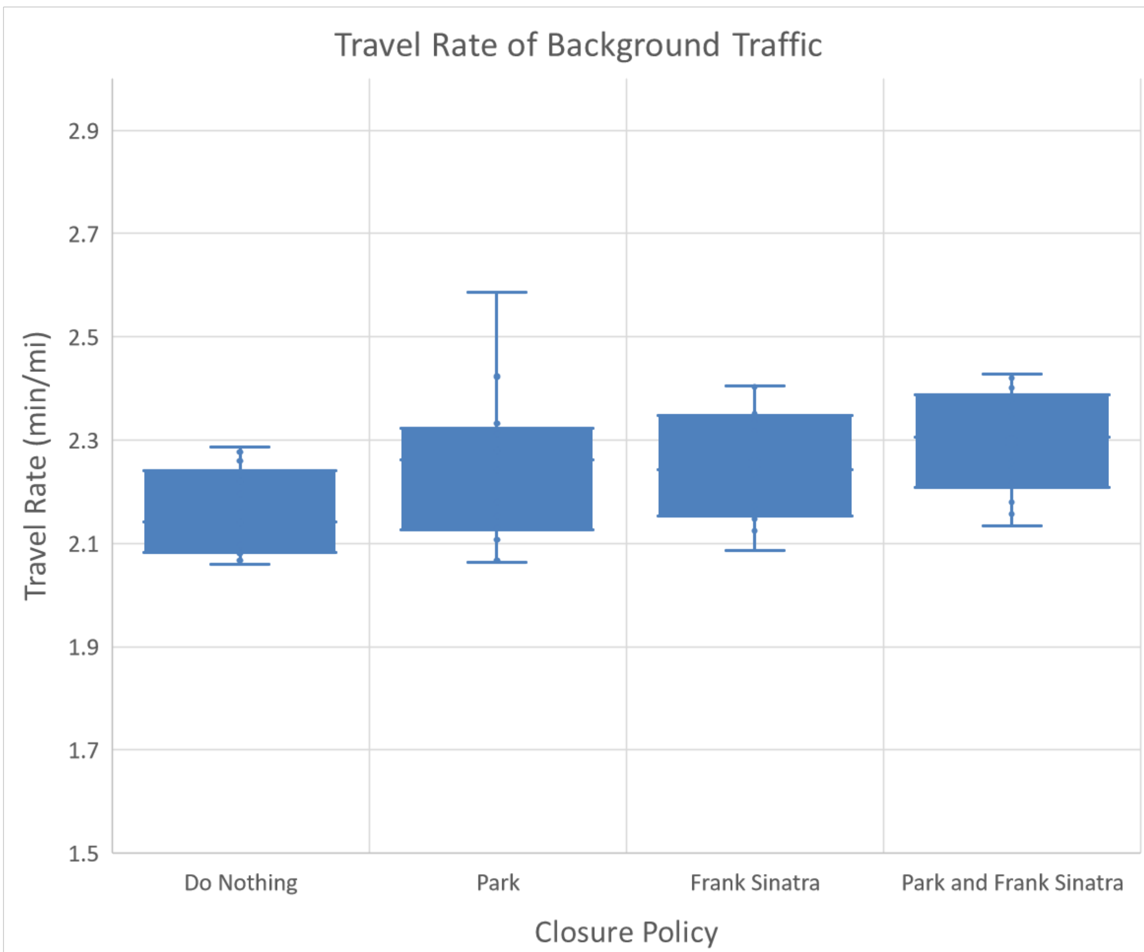


Figure 23. Travel rate of background traffic as a function of road closure policy lever

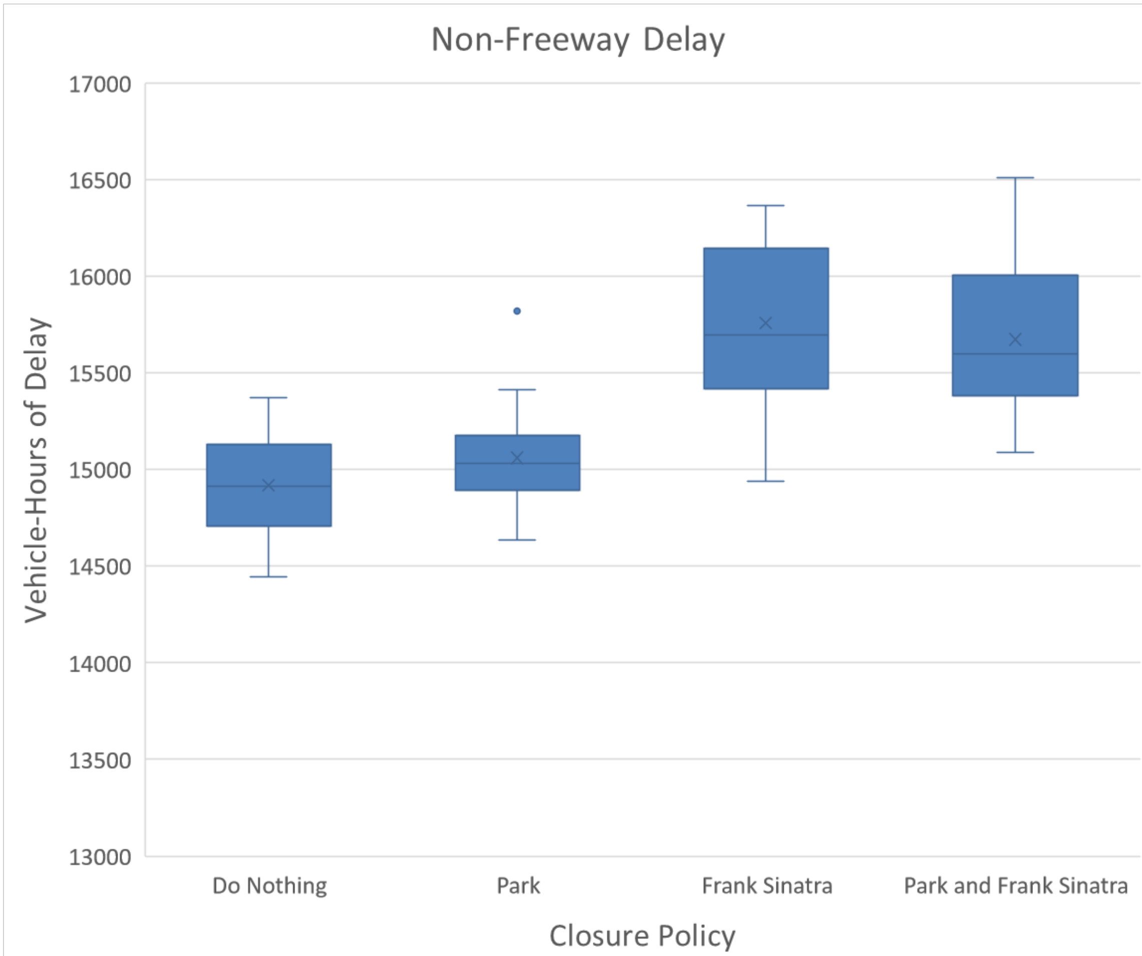


Figure 24. Non-freeway delay as a function of road closure policy lever

However, the travel rates and delays are only marginally greater when Park Ave alone is closed than when no roads are closed (i.e., the "Do Nothing" closure policy). Hence, it may be deduced from these experiments that closing Park Ave is a viable policy if reasonable pedestrian safety and mode shift benefits, which were not accounted for in these experiments, are expected. Similarly, it may be concluded that closing Frank Sinatra Dr is not tenable. This makes sense because a heavily traveled Las Vegas Blvd would probably have to bear the additional traffic volumes as the principal alternative north-south arterial to Frank Sinatra Dr.

While game attendance and road closures are clear factors in determining performance, it is interesting to note from review of the EMAT dashboard visualizations and from more thorough exploration of the raw performance measures that the two policy levers aimed at mitigating the game attendance and road closure impacts are largely ineffectual. That said, it can just narrowly be seen in Figure 26 that the effects of ramp metering have the expected directional influence, though the magnitude would render that influence negligible. When the Tropicana Ave-I15 ramp meters are shut off at 5:00 PM, there is a slight drop in surface street (i.e., non-freeway) delay and a slight uptick in freeway delay relative to when ramp meters are allowed to remain on throughout the PM peak. This is the expected trend, as ramp metering is designed to moderate traffic from the entrance ramp in the merging area, thus causing traffic to wait in queue longer on the surface streets before entering I-15.

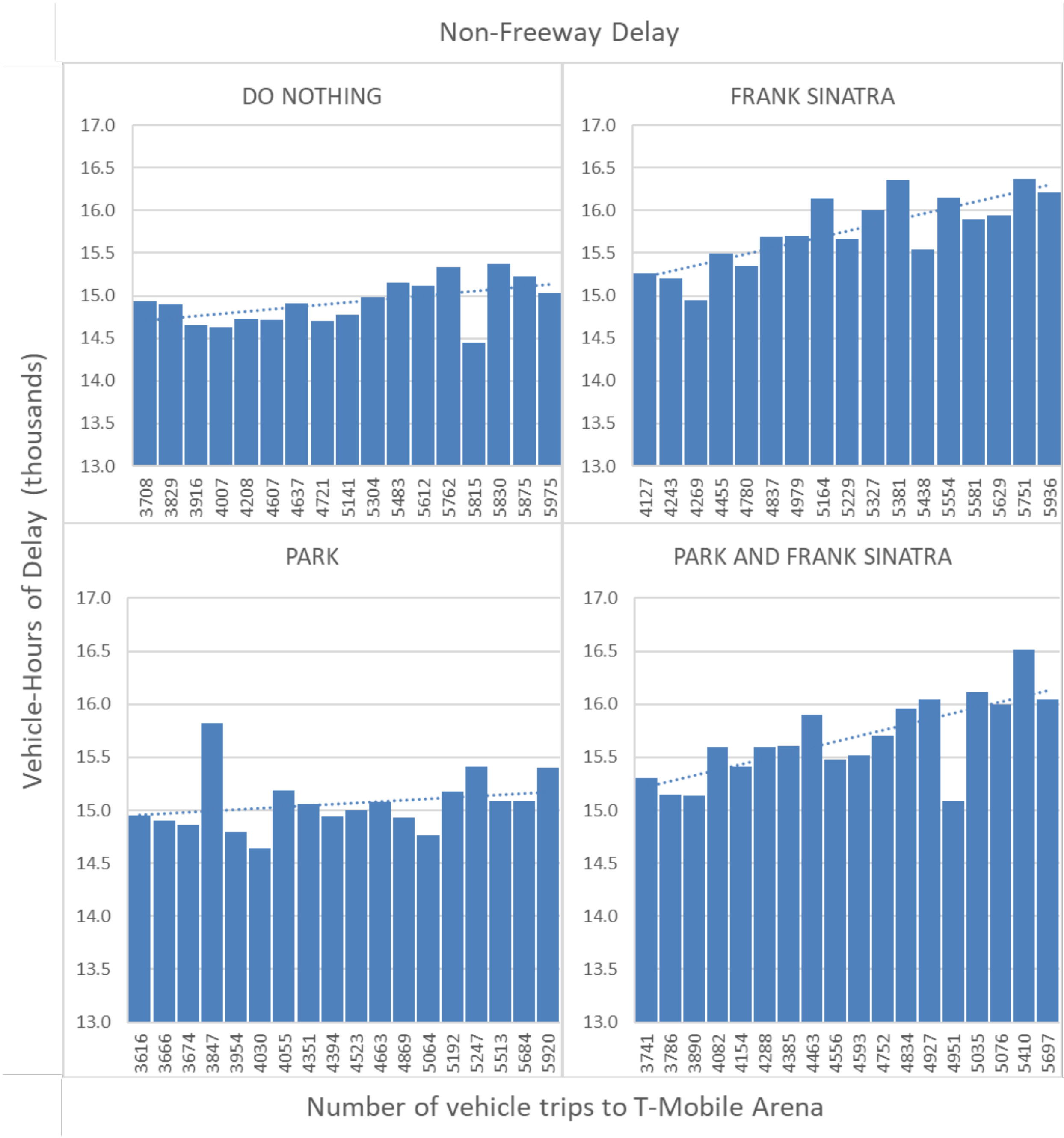


Figure 25. Travel rate of background traffic as game attendance increases under various road closure policies

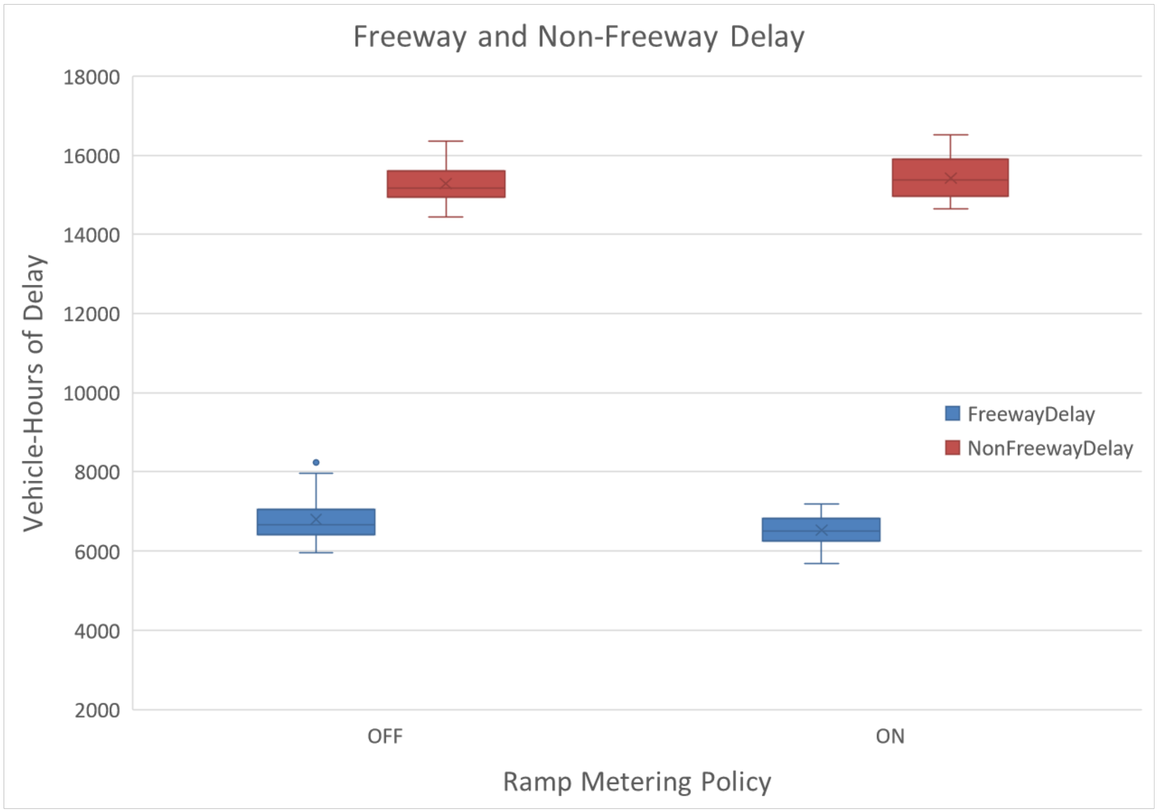


Figure 26. Freeway and non-freeway delay as a function of ramp metering policy

However, while the core model appears to have the appropriate sensitivity to the ramp metering policy lever, ramp metering does little to mitigate the adverse effects of game traffic and road closures. This leaves the DMS policy as the last resort. While the simple policy lever whether to employ DMSs or not has no discernible influence over outcomes, the DMS compliance uncertainty appears from the feature score values in Figure 19 to have influence over freeway delay, the travel rate of background traffic, and VHT that is commensurate with game attendance. However, no trend was evident in the data even when the experiments in which DMSs were in use were explored in isolation from those in which DMSs were not used.

In summary, the experiments run with the microsimulation-based DTA were useful in differentiating road closure policies that may be viable from those that may be problematic, and the potential impacts of these policies alongside game attendance could be quantified. However, none of the mitigation policy levers that were explored were determined to be helpful in mitigating increased delays resulting from game traffic.

6.0 EXPLORATORY ANALYSIS WITH MESOSCOPIC DTA

The second exploratory analysis with DTA used a mesoscopic simulation-based DTA called TransDNA, which performs dynamic traffic assignment (DTA) on large-scale, regional networks in TransCAD. TransDNA takes account of the geographic and geometric attributes of the planning network in the DTA, including intersection geometry and traffic signal timings that play crucial roles in determining interactions between time-varying travel demand and roadway capacities.

TransDNA uses time-varying origin-destination (O-D) demand input and propagates vehicle trajectories based on a mesoscopic traffic simulation engine. The mesoscopic simulator uses speed-density relationships and roadway capacities to capture congestion effects, delays, and the formation and dissipation of queues and spillback effects. The link travel times from the mesoscopic simulation are used to compute dynamic equilibrium solutions that are consistent with the simulated drivers' time-varying route choice decisions. Network conditions are displayed in dynamic, color-coded maps, where users can choose the measure of performance to view (e.g., density, flow, or speed). Thus, the dynamic pattern of traffic flow can be observed and monitored by iteration as the DTA progresses without the need for post-processing.

To demonstrate another way in which DTA can be used with EMA, the experiments were run as one-shot simulations with the input historical travel times derived from a DTA. In some EMA applications, it may be sufficient to run the DTA to convergence outside of EMA and then to run one-shot simulations in each experiment. This approach is reasonable when it can be assumed that travelers would have no opportunity to change their route in response to incidents or events that might change the level of service they experience. It can be argued that many travelers either may not know about the VGK game in advance or would not have enough experience driving in game-day conditions to know which routes may be preferred.

Hence, the experiments run with TransDNA are different from those that were run using the microsimulation-based DTA in TransModeler, wherein a DTA was run for each experiment. Thus, comparing the two sets of experiments might give an indication of how the results differ under each assumption.

The uncertainty variables and policy levers chosen for analysis with the mesoscopic DTA are similar to those chosen for the experiments with the microsimulation-based DTA. For discussion of those uncertainty variables and policy levers, please refer to 4.0 Exploratory Analysis with Microsimulation-based DTA.

Uncertainty Variables

A subset of the uncertainty variables and policy levers used in the microsimulation-based DTA experiments were used with the mesoscopic DTA. This is because mesoscopic simulation does not simulate traffic with the same level of detail and fidelity as microsimulation and thus does not support all of the same dimensions of uncertainty. In the experiments with the mesoscopic DTA, the lone uncertainty variable was game attendance, or the number of auto trips attracted to T-Mobile Arena for the VGK game.

The variable *Attendance* represents the total number of vehicle trips to the game and captures the uncertainty surrounding the increased traffic in the vicinity of T-Mobile Arena during the PM peak period. As with the microsimulation-based DTA experiments, the additional trips to T-Mobile Arena were assumed to have the same origin distribution as the base case. The trips were uniformly distributed between 4:30 PM-6:00 PM —from 1.5 hours before the game start time. A separate game trip matrix was created to model this demand which is scaled as per the value of the uncertainty variable — *Attendance* — in the specific EMAT experiment.

Policy Levers

The policy levers used in the microsimulation-based DTA experiments and that can be represented in the mesoscopic simulation include:

1. Road Closure
2. Ramp Metering at Tropicana

Road Closure

Four road closure policies were considered:

1. Do Nothing: No roads are closed.
2. Park: Segments of Park Ave between Frank Sinatra Dr in the west and Las Vegas Blvd to the east are closed to all vehicle traffic.
3. Frank Sinatra: The segment of Frank Sinatra Dr between Park Ave in the north and Arena Dr to the south is closed to all vehicle traffic.
4. Park and Frank Sinatra: All segments of Park Ave and Frank Sinatra Dr described above are closed to all vehicle traffic.

Ramp Metering at Tropicana

The ramp metering policy lever took on one of two categorical values in the EMAT experiments: ON or OFF. When ramp metering is ON, we reduce the capacity from 1800 vph to 1440 vph. When ramp metering is OFF, the capacity of the ramp is set at 1800 vph.

Exploratory Scope and Performance Measures

In the exploratory scope, the distributions of the uncertainties described above are specified. Policy levers and their values are also specified. The uncertainty distributions employed are summarized in Table 7, and the policy levers and their values are listed in Table 8.

Table 7: Uncertainty characterization for RTC EMAT analysis with the mesoscopic DTA as core model

Factor	Distribution	Minimum	Maximum
Attendance	uniform	2600	6000

Table 8. Policy levers for the RTC EMAT analysis with the mesoscopic DTA as core model

Factor	Values
Closure	Do Nothing / Park / Frank Sinatra / Park and Frank Sinatra
RampMetering	ON / OFF

Table 9 presents the performance measures from the core model that were chosen for the EMAT analysis. Scripts were written in GISDK to capture these performance measures from the raw simulation outputs that are meaningful in the context of the VGK game scenario. All performance measures are summarized from the entire period of simulation between 1:00 PM and 6:00 PM — that is the entire PM scenario period. Figure 28 presents the subset of links —called the Arena links— in the vicinity of the arena that are likely to be directly affected by the game traffic.

Table 9: Performance measures for RTC EMAT analysis with the macroscopic DTA as core model

Measure name	Geography	Trips	Description
Total_VMT	Entire model area	All trips	Vehicle miles traveled
Total_VHT	Entire model area	All trips	Vehicle hours traveled
Arena_VMT	Arena subarea	All trips traveling in subarea	Vehicle miles traveled
Arena_VHT	Arena subarea	All trips traveling in subarea	Vehicle hours traveled
CompletedTrips	Entire model area	All trips traveling in subarea	Total number of trips successfully completed
Arena_Freeway_VDT	On the freeway system (primarily I-15), including ramps	All trips	Vehicle hours of delay experienced on freeways and ramps
Arena_NonFreeway_VDT	On surface streets/ All trips	All trips	Vehicle hours of delay experienced on surface streets
Background_TravelRate	Entire model area	Background trips	Minutes per mile traveled
Game_TravelRate	Entire model area	Game trips to T-Mobile Arena	Minutes per mile traveled

Figure 27: The network with Arena Links

Exploratory Analysis and Results

As with the microsimulation-based DTA experiments, the results of the experiments are analyzed using scatter plots and feature scores. Specifically, the analysis is aimed at identifying the policies that would improve the performance measures across the uncertainties. Given the limited number of uncertainties and policy levers a total of 60 experiments were run in TransDNA.

Scatter Plots

Figure 29 presents the scatter plots of the VMT and VHT in the complete region and in the Arena region. The policy lever *Closure* seems to have a significant effect on the Total region's VMT and VHT; the effect is more apparent when we consider only the VMT and VHT in the subarea around the arena. Specifically, the closure of Park Avenue links reduces the Arena VMT but increases the Arena VHT. This suggests that closing Park Avenue leads to significant increases in delay and a consequent reduction in the number of trips that successfully reach their destination in the analysis period. As with the microsimulation-based DTA experiments, the *RampMetering* policy appears to have limited effect on VMT and VHT, both regionally and near the arena.

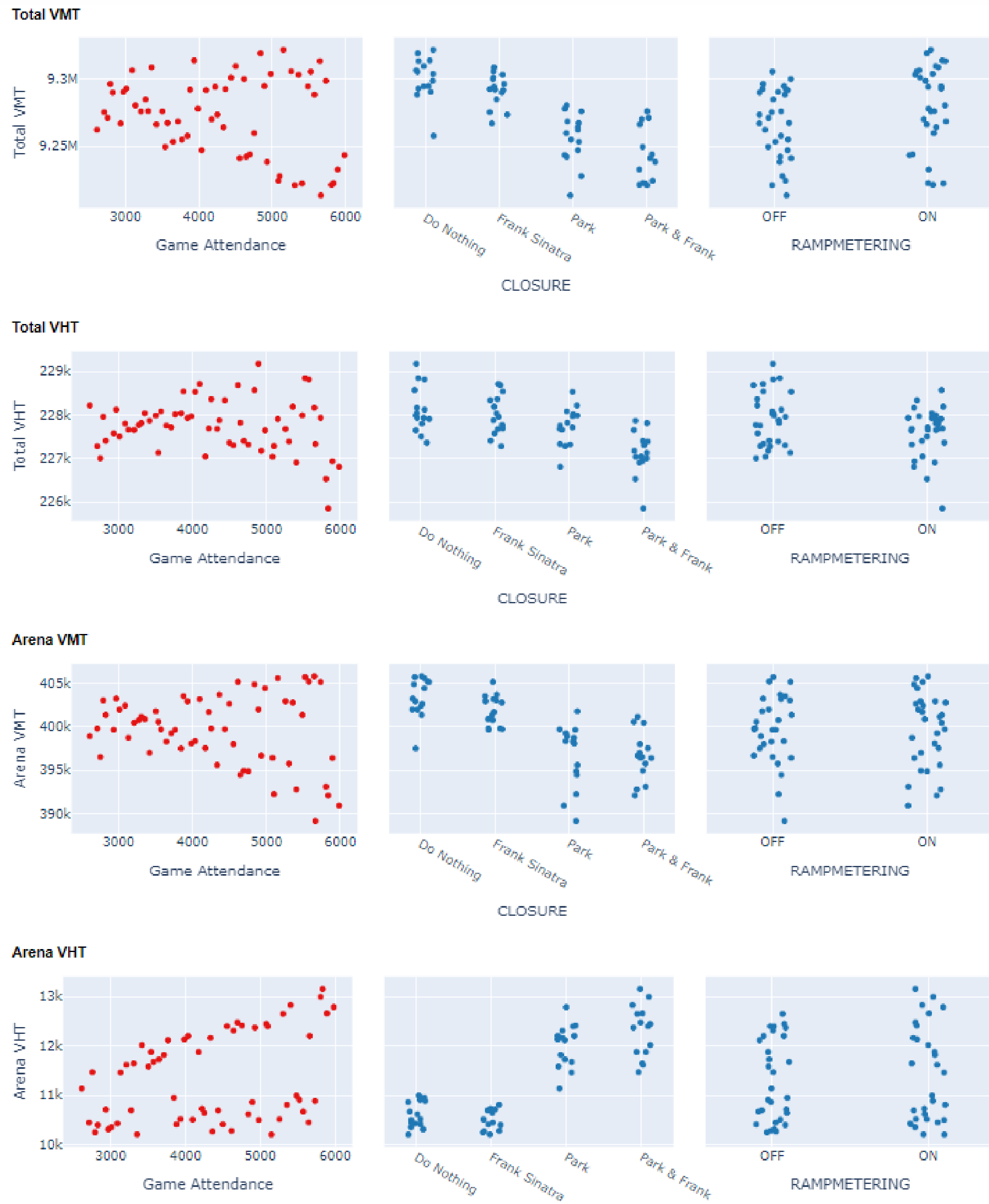


Figure 28: Scatterplots of VMT and VHT with respect to uncertainties and policy levers in TransDNA experiments

Interestingly, the effect of *Attendance* on VMT and VHT appears to depend on the policy levers associated with the experiment. Figure 30 presents the scatter plots of VMT and VHT with respect to game attendance differentiating different levels of *Closure*. Specifically, when Park Avenue links are closed, an increase in game attendance results in decrease of VMT and an increase in VHT. This is the result of fewer trips reaching their destination because of congestion and increased time spent in travel. This effect is more pronounced in the Arena subarea, confirming the expectation that the impacts of the road closure policy are most directly felt in close proximity to the arena.

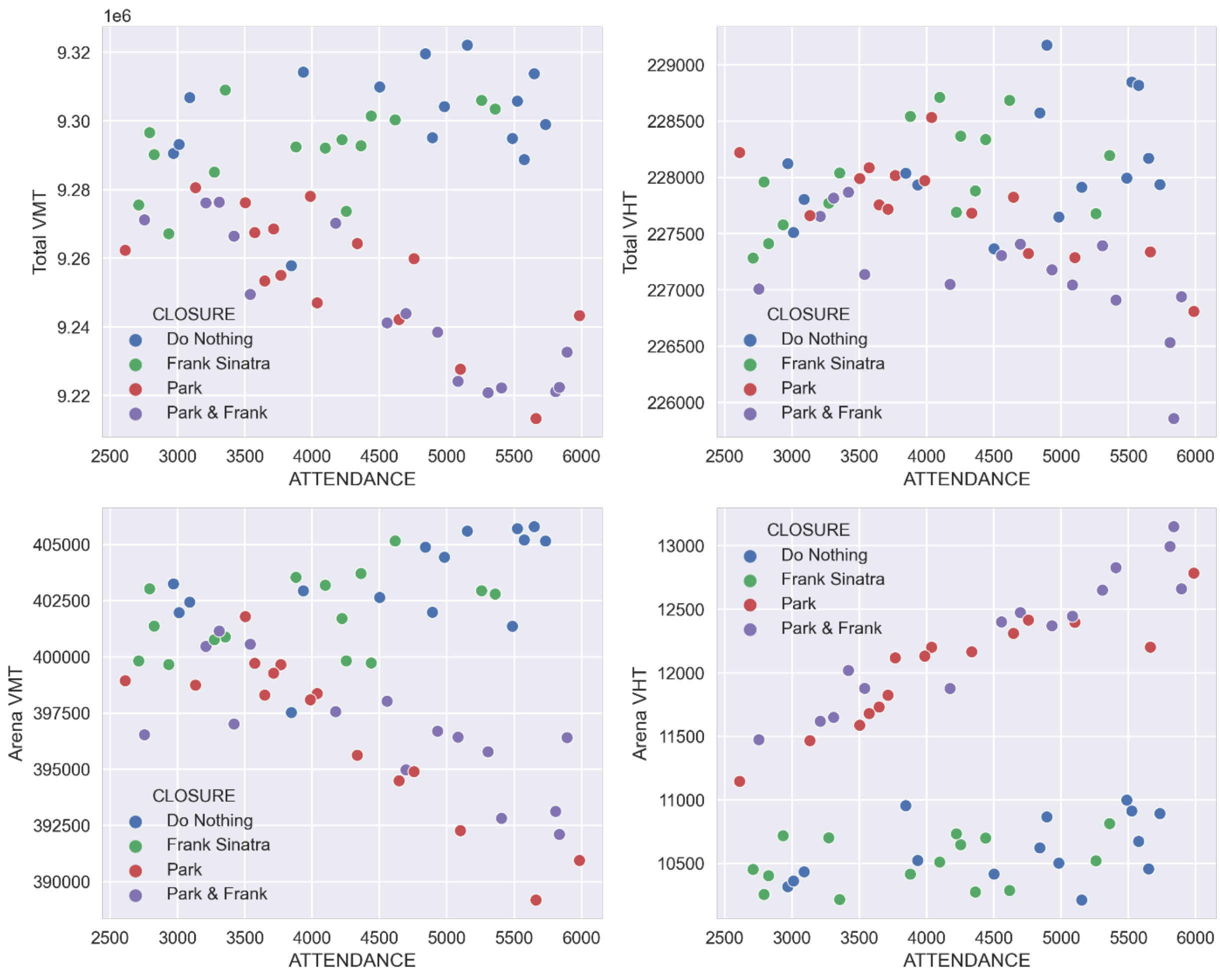


Figure 29: Relationship between Game Attendance and VMT, VHT in TransDNA experiments

Figure 31 depicts the scatter plots of the delays in the Arena subarea with respect to the uncertainties and policy levers. From the plots, the freeway delay is only marginally affected by the uncertainties and policy levers. On the other hand, delay on non-freeway links increases when Park Avenue links are closed and when ramp metering is ON. From the Figure 32, we can see that it is the closure of Park Avenue links that exacerbate the effect of an increase in game attendance on the non-freeway delay.

Figure 30: Scatter plots of delays in Arena links with respect to uncertainties and policy levers

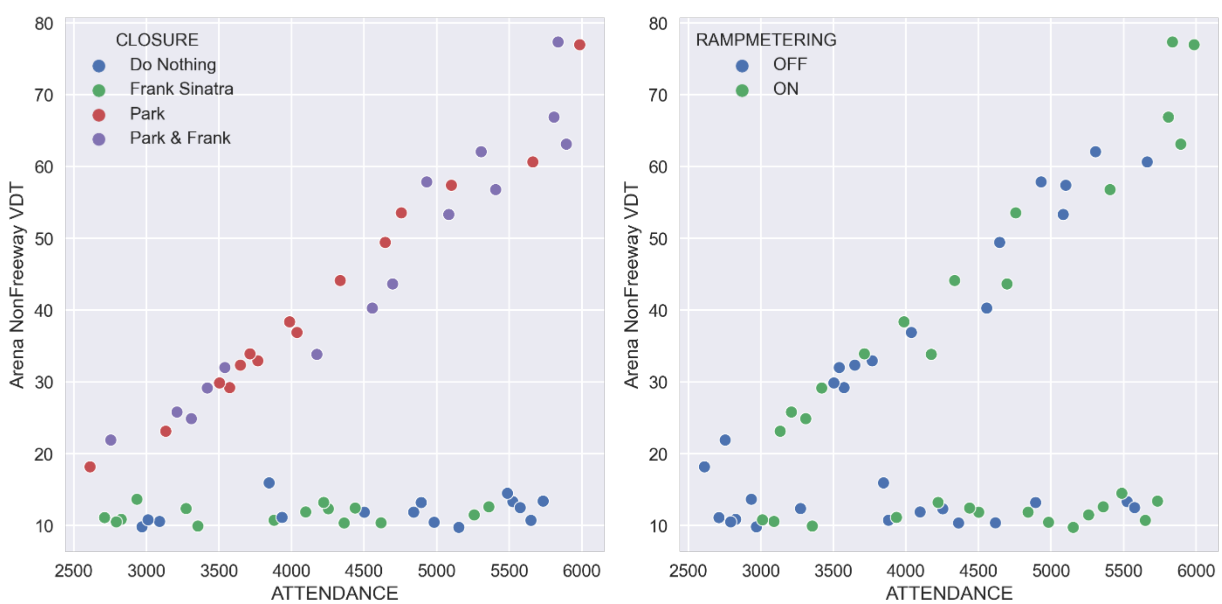


Figure 31: Relationship between delay on Nonfreeway links in Arena and game attendance in TransDNA experiments

Figure 33 presents the scatter plots of travel rates with respect to the uncertainties and policy levers. As before, we see that closing Park Avenue links increases the congestion and therefore the travel rate; however, this effect is limited to game trips. Further, ramp metering benefits the background traffic by reducing its travel rate. In other words, shutting off ramp metering penalizes background traffic while providing no measurable benefit to game trips.

Figure 32: Scatter plots of travel rates with respect to uncertainties and policy levers in TransDNA experiments

Feature scores

Figure 34 is a visualization of the features scores for all output measures. For a given output measure, the important (i.e., most influential) input variables are highlighted. It can be seen that the policy level *Closure* is the most influential with respect to performance measures region-wide VMT, Arena VHT, non-freeway delay, and Game traffic travel rate. *Attendance* is the most influential with respect to region-wide VHT and freeway delay. Finally, Ramp metering affects the background traffic's travel rate the most. These values reflect the observations from the scatter plots above.

From threshold feature scores in Figure 35, *Attendance* influences game travel rate when it is very low and very high. However, for moderate game travel rates it is the *Closure* that holds the most explanatory power. The background traffic's travel rate does not vary much, but the influence of the *Ramp Metering* policy on its value can be seen.

	ATTENDANCE	CLOSURE	RAMPMETERING
Total_VMT	0.325154	0.579974	0.094872
Total_VHT	0.437547	0.452331	0.110123
Arena_VMT	0.408274	0.563555	0.028171
Arena_VHT	0.282977	0.685914	0.031109
Arena_Freeway_VDT	0.633276	0.283213	0.083512
Arena_NonFreeway_VDT	0.394588	0.577250	0.028162
CompletedTrips	0.294407	0.624937	0.080656
Game_TravelRate	0.300504	0.668710	0.030786
Background_TravelRate	0.376992	0.182771	0.440237

Figure 33: Feature scores for output variables in TransDNA experiments

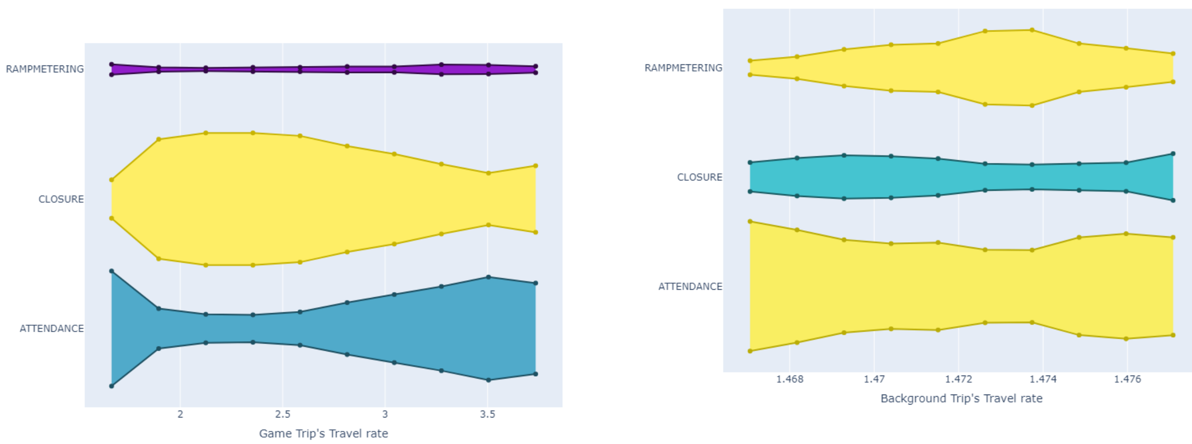


Figure 34: Threshold feature scores of travel rates in TransDNA experiments

In conclusion, closing Park Avenue links is likely to worsen the performance measures related to game traffic and in the vicinity of the arena. This effect is exacerbated when the number of game trips is high. Shutting off ramp metering offers no mitigating benefit for game trips and likely serves only to disadvantage background traffic. The analysis indicates that additional performance measures, such as those relating to pedestrian safety or transportation network company (TNC) services, might be useful to better gauge the performance of the policy levers. Further, additional policies, such as specific TNC drop-off locations and managed traffic diversion, may be more effective in mitigating congestion arising from game traffic.

7.0 LESSONS LEARNED

The software tools developed as part of this project place EMA in the RTC toolbox and TMIP-EMAT more specifically at the fingertips of the RTC's transportation modelers. The integration of EMAT with TransCAD and TransModeler makes all the provisions necessary for performing EMA with EMAT from within the graphical user interface in either product. While the application that was developed facilitates all of the mechanics of EMA, it does not, nor can it, distill EMA into a simple drag-and-drop, point-and-click procedure. Successful EMA application will continue to and for the foreseeable future hinge upon:

1. Sound experimental design
2. Some measure of proficiency with the preexisting APIs in TransCAD and TransModeler
3. Accommodation for lengthy model running times

Sound Experimental Design

Sound experimental design itself entails selection of uncertainty variables and policy levers that are supported by the core model and to which the core model is appropriately sensitive. If an uncertainty variable of concern or policy lever of interest is not already a parameter of, nor input to, the core model, then its evaluation may be a non-starter.

For example, in the stakeholder outreach phase of this project, the urban growth boundary (UGB) was proposed as a candidate uncertainty variable. This was an astute recommendation: the geography of the UGB will have an important effect on the geospatial character of land use and the urban and suburban populations and, ultimately, on a great many transportation planning performance measures. Hence, the UGB is a critical source of uncertainty and an entirely reasonable recommendation. However, land use forecasts are external to the TDM and are developed independently by various local governments in the region. The UGB is not a direct input to the TDM. If the UGB is redrawn, the TDM's predictions do not change unless the land use forecasts are remade in a way that accounts properly for the change in the UGB. This doesn't entirely rule out UGB as an EMA factor if an alternate set of proposed UGBs can be identified and the consequent land use and demographic forecasts produced. However, this raises the barrier to EMA considerably.

While stakeholder involvement and buy-in is a key to successful EMA process, good stakeholder ideas are not necessarily actionable within the confines of the core model. Reducing the number of candidate uncertainty variables to those variables that are already model parameters or inputs may be overly limiting. In truth, any application for which EMA is a candidate is likely to have a key set of potential uncertainty factors in mind at the outset. This is where proficiency with the APIs in TransCAD and TransModeler is pivotal: the core model may need to be changed to accommodate the experimental design.

In the examples described in this report, the core models were modified in various ways. For example:

1. To capture the rise of e-commerce as an uncertainty variable in the EMA performed with the RTC TDM in TransCAD, scaling factors were introduced that reduce the number of shopping trips and replace them with the truck trips.
2. To simulate vehicle trips to T-Mobile Arena for a VGK game in the DTA models, a separate trip matrix was created, and script was written to scale the volume of trips to be simulated from the matrix to account for uncertainty in attendance.

Sound experimental design also hinges on verifying that the core model has the appropriate sensitivity to the parameter or variable representing the uncertainty or policy lever. Hence sensitivity testing is advised as a preparatory step except where there is already a track record of the factor's influence over the model's performance measures. In the experiments described in this report, many experiments were run only to find that certain policy measures such as DMSs and ramp metering were ineffectual in mitigating the delays directly correlated with increasing VGK game attendance.

Sensitivity testing may also help to reveal when two factors have canceling effects (e.g., one contributes to increased delays while one contributes to diminishing delays). Absent sensitivity testing, the result of two variables canceling one another out may lead to the erroneous conclusion that the variables have no effect on the performance measures. It should also be noted when choosing uncertainty variables and policy levers that EMAT does not have a way of capturing correlations between factors. Rather, they are treated as independent effects.

In addition to sensitivity testing, it is advised that adjustments made to a core model to accommodate factors that were not previously a part of the core model structure be subject to calibration and validation prior to their use in EMA.

Proficiency with Core Model APIs

In some EMA applications, there may be some uncertainty variables that are represented by global model parameters (e.g., percentage of the region-wide vehicle fleet that are CAVs). These are the simplest to incorporate into EMA because the integration of EMAT with TransCAD and TransModeler allows for these parameters to be identified and adjusted by EMAT without any involvement of the TransCAD or TransModeler APIs.

However, comfort and proficiency with the APIs in TransCAD and TransModeler are likely to be necessary because most EMA applications will require application-specific customizations, which may include:

1. Simulation of a specific traffic control or management strategy.
2. Manipulation of the input traffic demand.

A key advantage of DTA over a TDM as a core model is the sensitivity to traffic operations that DTA affords. It is difficult to conceive of a DTA application in which there isn't a single policy lever that employs traffic control or traffic management in some fashion as a mitigation strategy.

While it is easier to imagine a DTA application in which the demand inputs remain fixed, a great many would require that the input demand be changed as part of the exploratory analysis, for example:

1. the demand for a special event,
2. the demand that will be attracted to a proposed land use development, or
3. the demand that would be induced when capacity is added.

Moreover, demand for events, new land uses, or new facilities is often quite difficult to predict, making it a prime candidate for inclusion as one of the key uncertainty variables.

If an EMA application with DTA is undertaken, and if the experimental design will explore traffic control or management policy levers, then the traffic control or management strategy is likely to be focused on a target location or facility. Hence, it will probably be necessary to manually prepare the model inputs reflecting those strategies in advance and to have EMAT invoke them through the use of categorical factors in the scope file. For example, the ramp metering strategy at Tropicana Ave described in this report took on two simple categorical values: ON or OFF.

Further, if the EMA application will explore demand uncertainties, then the adjustments to demand are likely to be made locally, affecting origins or destinations near a project or special event or affecting a segment of the driving public (e.g., identified by origin-and-destination pairs) most directly impacted by a project.

Because a special event or project will have impacts that have specialized, localized, or targeted impacts, the API in TransCAD and TransModeler, most often GISDK, will be needed to relate an uncertainty variable or policy lever in the scope file to an input specifically related to the project (e.g., a ramp metering strategy at Tropicana Ave or vehicle trips to T-Mobile Arena).

The same is true of performance measures: if the impacts are likely to be most felt in a particular area (e.g., an area of influence around T-Mobile Arena), on a particular facility (e.g., I-15), or by a particular segment of traveling public (e.g., fans driving to T-Mobile Arena), then it will be necessary to use the core model's API to extract the desired application-specific measure. The TransDNA and TransModeler DTA models produce a wealth of output data that can be summarized at any geographic or temporal scale or resolution.

The APIs in TransCAD and TransModeler may be needed for other reasons to support future EMA applications by the RTC. As the exploration of e-commerce in this report illustrates, the core models themselves may need to be amended or extended to incorporate the influence of a variable that does not have any representation in the model's existing structure. When the DTA in TransModeler is the core model, a wide range of possibilities leveraging the APIs in TransModeler emerge, including exploration of custom speed harmonization or queue detection algorithms that would dynamically invoke the DMSs to display reduced speed limits or queue warning messages.

Model Running Time Considerations

A key challenge in successful deployment of the EMA application developed in this project is making economical use of the core models. The TDM in TransCAD and the DTAs in TransDNA and TransModeler are computationally intensive models to runs. They take hours to complete one model run. Hence, dozens of EMA experiments can lead to model running times of days or possibly weeks depending on the number of factors in the experimental design. As mentioned previously, first subjecting the core models to calibration, validation, and sensitivity testing can avoid running experiments exploring factors that have no discernible influence on the performance measures, hence contributing to burdensome model running times while adding little informative value to the decision-making process.

Because long model running times are unavoidable up to a point, it is worth considering ways in which model running times can be reduced:

1. Is EMA a good fit for the analysis? If the question at hand can be answered by carefully choosing a few values for one or a small number of variables across the expected range of values and running the model for a manageable number of scenarios, then this more traditional approach to scenario analysis may be more suitable than full-fledged EMA.
2. If the core model is the TDM, do all steps of the model need to be performed in every experiment? If the first steps in the TDM are unaffected by the changing values of the experimental factors, then a scenario can be configured in which the model execution begins from a later step, the inputs to which are identical across all experiments.
3. If the core model is DTA, is DTA needed, or will a one-shot simulation suffice? For some applications, it can be argued that drivers do not have an opportunity to learn the new congestion patterns that develop under a given scenario. DTA emulates the process of learning recurring congestion patterns to which drivers adjust their route choices. In some applications, such as those involving unplanned disruptions or incidents, this opportunity would not exist, and a single simulation would be more appropriate.
4. If the core model is DTA, can the DTA be warm-started and run for fewer iterations? In most applications, it should be possible to begin the DTA experiments with the congested travel times and delays of a previously converged DTA run. If the policy levers or changes in demand are not a dramatic departure from that previous DTA, then it may be possible to run the DTA to convergence in many fewer than the typical 50 iterations that are advised. However, some experimentation may be needed to determine how many iterations are sufficient for a given application.
5. If the core model is DTA, is microsimulation required? Certain applications that require operational sensitivity to detailed traffic signal or ITS strategies will require the microsimulation-based DTA in TransModeler and hence will require longer running times. However, applications that do not call for this level of operational detail may enjoy the shorter running times of the mesoscopic DTA in TransDNA. This question goes back to sound experimental design: the selection of the core model should take into consideration the types of uncertainties that are to be explored.

When designing any EMA application, the express consideration of the questions above should be routine in the planning stage so that the most can be made of model running times.

Other Lessons Learned

Numerous occasions arose in the course of the project in which unanticipated difficulties were encountered working with EMAT itself. The difficulties arising from the use of EMAT had partly to do with the lack of continuous software documentation and support. The documentation regarding the installation and application of EMAT is well-written and well-organized, but the troubleshooting guidance was thin. Strong familiarity with Python on the project team was critical to the success of the experiments described in this report.

The problems that were encountered had most often to do with EMAT installation and incompatibilities that arose between the EMAT interfaces and the third-party libraries on which EMAT relies. When changes in Python and in the third-party libraries arise, the EMAT interfaces are susceptible to breakage.

Future applications should expect that similar difficulties may arise, and plans should be made to have the technical resources to overcome them. EMAT is an open-source product that relies on volunteer contributors. As such, there is no avenue for technical support. However, there are other peer agencies that use EMAT or have used it in the past. It may be worthwhile to form contacts with these agencies to learn from their experiences and to create a channel for collaboration when there is need for troubleshooting.

8.0 CONCLUSIONS

The software that was developed to integrate the RTC's planning and DTA models with EMAT greatly diminish the technical barriers to EMA. The experiences developing and applying a practical EMAT application described in this report demonstrate that successful EMA is now within the RTC's grasp. However, successful EMA will require more than the software tools that enable EMAT to leverage the RTC's planning and DTA models as core models. Successful implementation and deployment at the RTC will also require certain technical resources (e.g., in-house Python experience) and the schedule to accommodate potentially lengthy model running times. However, for those commitments, EMA can potentially provide valuable returns through better-informed, more robust decision-making.

Though the e-commerce, visitor, and VGK scenarios described in this report are intended simply to be demonstrative of EMA's potential, their example and the accompanying guidance in this report are designed to help the RTC navigate and apply EMA in future projects when faced with deep uncertainty.

The experiences documented in this report are also a potential resource to a broader transportation planning practice. They present a template for integrating EMA with DTA, a match not previously made in the research or practice. DTA, when combined with thoughtful experimental design characterized by a conservative approach to selecting uncertainty variables and policy levers and by careful consideration of running time reduction strategies, can be successfully deployed as a core model in EMA.

More importantly, the successful demonstration of EMA integrated with DTA is further evidence that EMA has much to offer transportation planners seeking a better way to manage uncertainty, prepare for disruptive technological change, and explore innovative policies and solutions.

APPENDIX A: USER'S GUIDE

This appendix gives instruction how to design and run EMA experiments with the EMAT application described in this report. In this appendix, you will find:

1. Installation instructions for EMAT.
2. Step-by-step instructions for running EMA in TransCAD or TransModeler.
3. A description of the GISDK interface with EMAT.

This appendix assumes familiarity with EMAT and EMA concepts. To learn more about EMAT, visit the TMIP-EMAT website:

https://tmip-emat.github.io/

Installation of Anaconda Python and EMAT

To run EMAT with Caliper platforms, the following installations and setup must be performed in order:

1. Close TransCAD or TransModeler. Before they can be used to interface with EMAT through Anaconda, they must be launched after EMAT has been installed.
2. Install EMAT (based on the TMIP-EMAT website):
    a. Install/update Anaconda
    b. Create the EMAT environment
    c. Install TMIP-EMAT within the EMAT environment
3. Install Python libraries required by Caliper
    a. Install the CairoSVG library

The above steps are detailed below.

Install Anaconda

If Anaconda is already installed on your computer, then refer to the EMAT website for instructions creating the EMAT environment:

Otherwise, install the latest Anaconda Individual Edition from the following link:

Anaconda | Individual Edition

Please note the following during installation:

- Choose to install Anaconda for "Just Me" rather than for all users, which will require administrator privileges and will preclude the option in the third bullet below.
- You may click Cancel once Anaconda is installed if you are prompted to install additional bundled tools such as PyCharm Pro. You may click Yes to confirm that you wish to quit the Anaconda3 Setup at that time.

- Also, check the box to have Anaconda added to the PATH environment variable. Anaconda warns against taking this step, but a conda.bat file installed by Anaconda must be found in the PATH variable for the EMAT integration to work in TransCAD or TransModeler. If the installation warning causes you concern, you may instead opt to add the path to the conda.bat file in your environment variable afterward. To do so, remember the install path that you have chosen for Anaconda (C:\Users\<username>\Anaconda3\ by default) and add the following path to the environment variable after installation is completed:

```
<AnacondaInstallPath>\Scripts\
```

Depending on the IT policies, it may not be possible to install the Anaconda according to the recommendations above. If Anaconda is installed for all users, then the conda command used by TransCAD and TransModeler to invoke EMAT will need to be specified when you configure EMAT experiments. Specifically, you will have to indicate the location and name of the conda batch file, typically named conda.bat and located in the following folder after installation:

```
C:\Users\<username>\anaconda3\Library\bin\
```

Create EMAT environment

After Anaconda has been installed, you will find the "Anaconda Prompt (Anaconda3)" app in the Windows Start menu. Find it and open an Anaconda Prompt command window.

In the Anaconda Prompt command window, execute the following command:

conda create --name EMAT "python=3.8.11"

Enter "y" for the prompts that follow. This will create a new Python environment variable named EMAT (please retain this name) and install Python version 3.8.11 in it. The interfacing of Caliper platforms with TMIP-EMAT has been tested on this version of Python, though the TMIP-EMAT website only requires version 3.7.

▶ Install TMIP-EMAT

TMIP-EMAT can be installed once the EMAT environment has been created by following the steps below. For more information about installing TMIP-EMAT, you are encouraged to consult the website https://tmip-emat.github.io/source/emat.install.html.

1. Open an Anaconda Prompt command (or use one that is already open).
2. Execute the following command to activate the EMAT Python environment:

   conda activate EMAT

3. Execute the following command to install TMIP-EMAT within the EMAT environment:

   conda install emat -c tmip -c defaults -c conda-forge

Enter "y" for the prompts that follow.

▶ Install CairoSVG library

A library called CairoSVG is required to support the interface between Caliper products and the TMIP-EMAT Python libraries and functions. Install the library using the following steps:

1. Open a command window with Administrator rights (or use one that is already open).
2. Execute the following command to activate the EMAT Python environment (if it has not been done already):

   conda activate EMAT

3. Execute the following command to install TMIP-EMAT within the EMAT environment:

   conda install -c conda-forge cairosvg

Enter "y" for the prompts that follow. Note that the CairoSVG library can technically also be installed using the pip package management system, but it is not recommended because it does not appear to install all the required dependencies.

How to Perform EMA with the EMAT Application

To perform EMA in TransCAD or TransModeler, you will first need to identify the uncertainty variables and policy levers you wish to explore and the performance measures you will use to evaluate them. This step may require the input and feedback of stakeholders. When the uncertainty variables, policy levers, and performance measures are decided, the following steps can be taken to perform EMA with the EMAT application:

1. In a text editor, create a scope file listing the uncertainty variables, policy levers, and performance measures.
2. Prepare the script in GISDK that will map uncertainty variables or policy levers named in the scope file to inputs into the core travel model or DTA model.
3. Prepare the script in GISDK that will extract performance measures from the core model's outputs.
4. Open the *EMAT Tools* toolbar in TransCAD or TransModeler.
5. Use the *EMAT Tools* to choose the scope file, create the experiments, run the experiments, and visualize the performance measures.

STEP 1: Prepare the Scope File

The scope file has a simple human-readable format called YAML. More information about YAML and what the scope file's contents should be to define your EMA, visit the EMAT website. The contents of the scope file for the TransDNA DTA experiments are provided below as an example.

```
# TransDNA DTA Scope File

scope:
    name: TransDNA DTA
    desc: Mesoscopic traffic simulation
```

```yaml
inputs:

    ATTENDANCE:
        ptype: uncertainty
        desc: Vehicle trips to Aria and New York-New York parking
        shortname: Game Attendance
        dtype: integer
        default: 4800
        min: 2600
        max: 6000
        dist: uniform

    CLOSURE:
        ptype: policy lever
        desc: Road Closure
        dtype: cat
        default: Do Nothing
        values:
            - Do Nothing
            - Park
            - Frank Sinatra
            - Park and Frank Sinatra

    RAMPMETERING:
        ptype: policy lever
        desc: Ramp Metering at Tropicana
        dtype: cat
        default: "ON"
        values:
            - "ON" #1000
            - "OFF" #1800

outputs:

    Total_VMT:
        shortname: Total Vehicle Miles Traveled

    Total_VHT:
        shortname: Total Vehicle Hours Traveled

    Arena_VMT:
        shortname: Vehicle Miles Traveled in Arena Subarea

    Arena_VHT:
        shortname: Total Vehicle Hours Traveled in Arena Subarea

    Arena_Freeway_VDT:
        shortname: Freeway Delay

    Arena_NonFreeway_VDT:
        shortname: Non-Freeway Delay

    CompletedTrips:
        shortname: Completed Trips

    Game_TravelRate:
        shortname: Game Trip's Travel Rate

    Background_TravelRate:
        shortname:  Background Trip's Travel Rate
...
```

The file is divided into three sections, each preceded by a line with the word scope, inputs, or outputs followed by a colon. In the scope section, a few attributes (e.g., name and desc, short for description), also preceded by colons, describe the analysis.

In the inputs section, uncertainty variables and policy levers, which are identified as one or the other by the attribute ptype, are listed. Additional information describing each factor, such as the data type (dtype), distribution parameters of numerical factors (e.g., dist, min, max), and list of values that a categorical variable can assume (values), is provided on lines indented beneath the factor name. Factor names (e.g., *ATTENDANCE*) are supplied by the analyst and will be mapped to core model inputs in Step 2.

Similarly, in the outputs section, performance measures are defined, followed immediately by attributes that describe them (e.g., shortname). Performance measure names are supplied by the analyst and will be mapped to core model outputs in Step 2.

STEP 2: Prepare Script to Map Exploratory Factors to Model Inputs

A GISDK resource file (.RSC) implements the translation of the EMAT uncertainty parameters and policy lever variables and their values as specified in the scope file (EMAT\config.yaml) to core model inputs and parameter values. The same resource file may also implement the post-processing and calculation of core model outputs, relating the output measures identified in the scope file by name to the outputs produced by the model.

A UI database is the resource file's compiled form and must be present before EMA can be run. Unless the contents of the resource file are changed, the UI database need only be compiled once. When the script is completed, compile the variable mapper resource file (e.g., VarMapper_UI.rsc) to a UI database (e.g., VarMapper_UI.dbd). The macro names implemented in the resource file and the UI database in which they are compiled will be identified with the *EMAT Tools* toolbar in TransCAD or TransModeler in Step 3.

An example of the macro that relates uncertainty factors or policy levers to model parameters or inputs, heretofore referred to as the parameters macro, is provided below. The parameters macro is called before each experiment is run to obtain an options array codifying the action to be taken to transform exploratory factor values into model parameters or inputs. Your parameters macro should follow a similar format, taking three arguments and returning an options array.

Per the example, the three arguments to the parameters macro are:

1. ExperimentVariable is the first argument and will take the string name of the input factor (e.g., “RAMPMETERING”).
2. VariableValue will take the value of the variable for this experiment (e.g., “ON”).
3. Opts is an options array that is only populated for TransDNA experiments. It has three options named “NetworkHandle,” “LineLayer,” and “NodeLayer” whose values are the handle to the network object used by the TransDNA project, the line layer representing streets in the TransDNA model, and the corresponding node layer, respectively.

The values that should be included in the options array that you return are:

1. “Method” and “Values,“ where Method is the string name of the method in the TransDNARunner or RunManager GISDK class in TransDNA or TransModeler, respectively, and Values is value of the argument that should be passed to the Method. In the example provided, *TransDNARunner* has a method named *SetCapacityPerLane* that takes an array of the field names that contain the per-lane capacity that are inputs to TransDNA in the *TransDNA Scenario Manager*.
2. “MacroName” and “Values,” where MacroName is the name of a macro in the variable mapper resource file that should be called before the experiment is run, and Values is an options array that will be passed to the macro as its sole argument.
3. “FileName” and either “TableName,” “Row,” and “Column” or “ItemName,” where FileName is the name of either the parameter file or road class file in which a model or road class parameter is found, and the other options describe the table and cell position or item name of a parameter whose value should be changed.

```
Macro "VariableMapper" (ExperimentVariable, VariableValue, Opts)
    validate = CreateObject("Caliper.Validate")
    ExperimentVariable = validate.GetString(ExperimentVariable)

    if ExperimentVariable = null or TypeOf(ExperimentVariable) <> "string" then
        Throw("Invalid experiment variable")
    else if ExperimentVariable = "RAMPMETERING" then do
        ret.Method = "SetCapacityPerLane"
        if VariableValue = "ON" then
            ret.Values = {"Capacity_metering"}
        if VariableValue = "OFF" then
            ret.Values = {"Capacity"}
    end
    else if ExperimentVariable = "CLOSURE" then do
        ret.MacroName = "Do Road Closure"
        ret.Values = {'VariableValue': VariableValue} + Opts
    end
    else if ExperimentVariable = "ATTENDANCE" then do
        ret.MacroName = "Do Game Demand Scaling"
        ret.Values = {'VariableValue': VariableValue}
    end
    else
        Throw("Experiment variable: '" + ExperimentVariable + "' not defined")

    return(ret)
EndMacro
```

When you use a macro (i.e., MacroName) to map the variable values to model parameters or inputs, you will need to write a macro that interprets the value for the factor and translates that value to action. The entire power of the GISDK in TransCAD and TransModeler is at your disposal. Hence, what you can do with the macro approach is limitless.

Below is an example from the TransDNA experiments. This macro “Do Road Closure,” which is the value assigned to MacroName when the ExperimentValue has the value “CLOSURE” in the variable mapper code example above, performs the tasks that are needed to effectuate the road closure policy for the experiment that is about to be run. This macro calls another macro “Get Links to Close,” which is omitted here for brevity but can be reviewed with the work products delivered with this report, to get a string query (closeqry) that can be used to select the links that should be closed based. In that macro, the “CLOSURE” policy lever passed to “Do Road Closure” is used to construct the query string to select the links matching the policy. Then, the GISDK class “Network.Update” and its method “DisableLinks” are invoked to disable the appropriate links. Lastly, the macro returns the name of a macro that should be called to undo or restore any changes that were made to the model inputs for the experiment. In this example, macro “RoadClosureCleanup” will restore the network to its original state, re-enabling the links that were disabled for the experiment.

```
Macro "Do Road Closure"(Opts)

    closeqry = RunMacro("Get Links to Close", Opts)

    neth = Opts.NetworkHandle
    netobj = CreateObject("Network.Update", {Network: neth})
    if closeqry <> null then do
        netobj.LinkLayer = Opts.LineLayer
        netobj.DisableLinks({Type: "BySet", Filter: closeqry})
        netobj.Run()
    end

    ni4 = GetNetworkInformation(neth)
    a = 1
    return("RoadClosureCleanUp")

EndMacro
```

TransCAD (or TransModeler, depending on the core model) takes the instruction to call macro “Do Road Closure” to effectuate the road closure policy before each experiment from the variable mapper macro “VariableMapper” and in turns take the instruction to call macro “RoadClosureCleanUp” when the experiment is completed from “Do Road Closure.”

Note that the macro that performs the tasks required by an uncertainty variable or policy lever is not required to return a macro name. If no cleanup task is needed to restore the model to its original pre-experiment state, then the macro need not return any value (i.e., no return() statement is needed). If a cleanup macro is used, then it should be implemented in the same resource file.

The third variable mapping option (i.e., FileName) allows you to specify a parameter among the built-in model or road class parameters in TransDNA or TransModeler that corresponds with the exploratory factor, typically an uncertainty variable. The TransModeler experiments described in this report used the CAV parameters in TransModeler to represent the CAV percentage uncertainty. In the variable mapper macro, the code implementing the translation looked like the following:

```
    else if ExperimentVariable = "CAV" then
       return({FileName: "pm.xml", TableName: "VehicleAutomationLevel", Row: 0,
               Column: 4})
```

The FileName references a TransModeler parameter file, and TableName references a table of parameters in that parameter file. Row and Column are 0-based indices to the cell in the table whose values should take on the value of the CAV uncertainty parameter. Row 0, Column 4 in table “VehicleAutomationLevel” in the model parameter file pm.xml in TransModeler refer to the cell highlighted in Figure 36 found when choosing *Parameters > Vehicle > Automation* from the menu system in TransModeler. Parameters that are not found in tables but may be represented by a single value can be referenced with ItemName rather than TableName. No Row nor Column need be specified when this is the case.

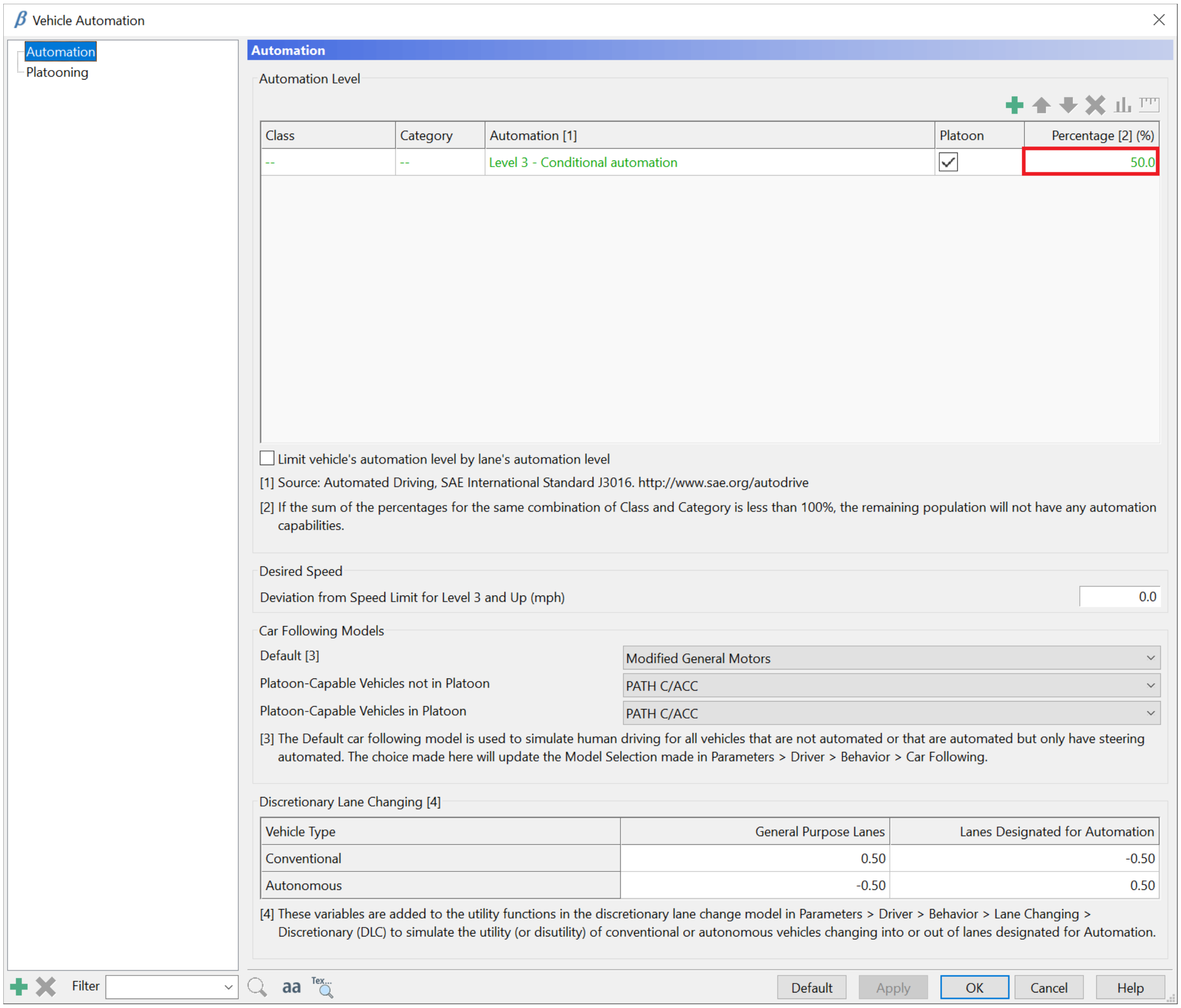


Figure 35. Location of the CAV percentage parameters in the TransModeler vehicle fleet parameters

As of the writing of this report, the FileName, ItemName, and TableName values are not published, but most parameters documented in TransDNA and TransModeler can be used. Caliper can provide the corresponding values of FileName, ItemName, and TableName upon request.

STEP 3: Prepare Script to Map Performance Measures to Model Outputs

Just as GISDK is used to write the parameters macro to map exploratory factors to model inputs, a *measures macro* should be implemented in the same resource file to retrieve the desired performance measures from the model’s outputs and to map them to the measures defined in the scope file by name. The measures macro takes one argument in DTA experiments with TransDNA or TransModeler: the file name of the trip table, which is a convenient source for numerous network-wide measures, such as VMT, VHT, vehicle hours of delay, and Completed Trips, all of which can be easily summarized from the table for all trips or for subsets (e.g., classes) of trips. The measures macro should return an options array in which the names of the options are the names of the performance measures in the scope file (e.g., “Total_VMT”) and the value is the value of the corresponding measure.

As with the macros that implement the changes to model inputs, the various GISDK classes in TransDNA and TransModeler, such as the TransDNARunner and the RunManager, respectively, will serve as powerful tools for post-processing raw DTA outputs and obtaining the exploratory measures.

STEP 4: Open the EMAT Tools

Open the *GISDK Toolbar* from the *Tools* menu and click *Immediate Execution* on the toolbar to open the *GISDK Immediate Execution* dialog box (Figure 36). In the dialog box, type the command RunMacro(“EMAT Tools”).

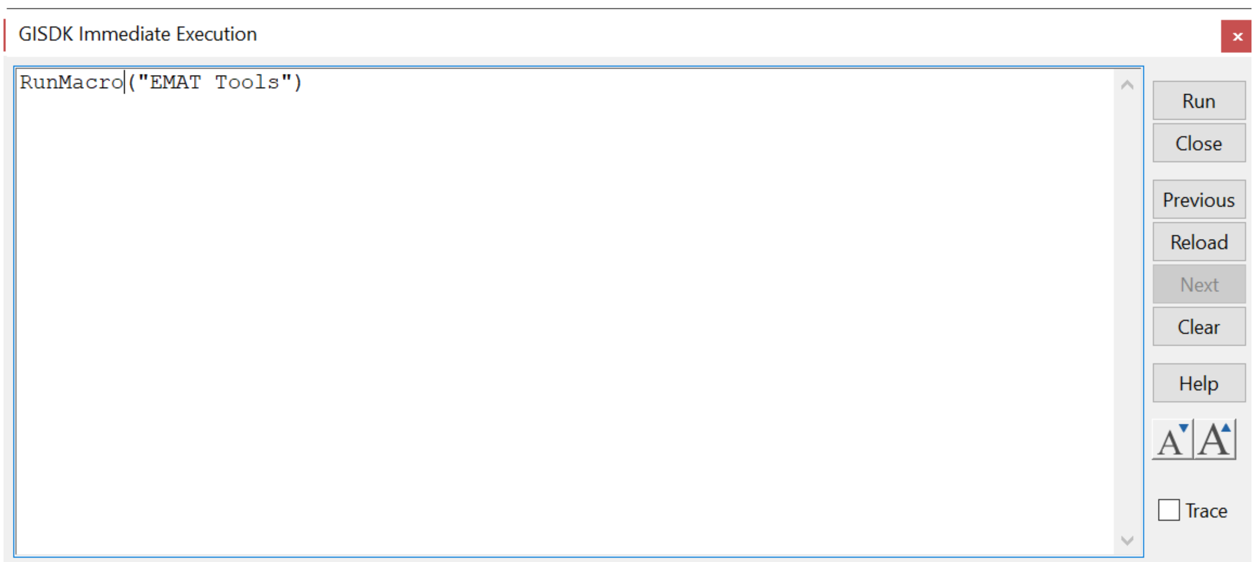


Figure 36. The GISDK Immediate Execution dialog box

Click *Run*. The *EMAT Tools* toolbar will open (Figure 37).

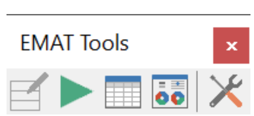


Figure 37. The EMAT Tools toolbar in TransCAD and TransModeler

STEP 5: Use the EMAT Tools in TransCAD or TransModeler to perform the EMA

The EMAT Tools toolbar has the following functions:

Button Name	Function
Create Experiment Table	Invoke EMAT to draw values of uncertainty variables and policy levers for all the experiments to be run and create a table with a record for each experiment and fields for each factor and performance measure. Enabled when the Settings button has been clicked and a scope file chosen. Disabled if the experiment table has already been created.
Run Experiments	Invoke the TDM or DTA model to run each of the experiments and populate the performance measures in the experiment table. Enabled when the Settings button has been clicked, a scope file chosen, and the experiment table created.
Open Experiment Table	Open the experiment table in a dataview. Enabled when the Settings button has been clicked, a scope file chosen, and the experiment table created.
Visualize Metrics	Open the *EMAT Metrics Visualizer* to choose the visualizations you wish to create from the performance measures. Enabled when the Settings button has been clicked, a scope file chosen, and the experiment table created, but will not produce useful visualizations until the experiments have been run.
Settings	Open the *EMAT Settings* dialog box to choose a scope file and identify the parameters and measures macros.

The first action you will take to set up your experiments in the toolbar will be to click the Settings button. The *EMAT Settings* dialog box will open (). You will need to do the following in the dialog box:

1. In the *Conda Environment* box, enter the name of the EMAT environment you used when you created the EMAT environment in Anaconda during EMAT installation.

2. If you did not install Anaconda for all users on the computer and/or did not choose to add Anaconda to the PATH environment variable during installation, then you will need to click the button to the right of the *Conda Path* box and browse for the batch file (typically C:\Users\<username>\anaconda3\Library\bin\conda.bat) where it was installed.

3. In the *Core Model* frame, the open simulation project will always be identified in TransModeler. In TransCAD a button will appear to the right of the *File Name* box. If TransDNA is the core model, click the button and choose the map file (.map) in which the TransDNA project is saved. If the TDM is the core model, click the button and choose the model file (.model) in which the flowchart model is saved.

4. In the *Scenario* drop-down list, choose the TransModeler project scenario, TransDNA scenario, or TDM flowchart scenario that should be run for the experiments.

5. If the TDM is the core model, choose the step that should be run from the *Run* drop-down list.

6. Click the button to the right of the *Scope File* box and choose the scope file (.yaml).

7. Click the button to the right of the *Experiment Table* box and choose the name of an experiment table (.bin) if one has already been created or enter a file name to create a new one.

8. In the *Number of Samples* box, enter the number of sample values to be drawn for each factor.

9. Click the button to the right of the *UI Database* box and choose the UI database (.dbd) into which the resource file containing the parameters and measures macros is compiled.

10. In the *Parameters Macro* box, enter the name of the parameters macro.

11. In the *Measures Macro* box, enter the name of the measures macro.

12. To save all of the settings entered above, click the *Settings* button and add a settings entry. In future sessions of TransCAD or TransModeler, you can click Settings and choose that entry to repopulate the dialog box from settings you previously saved.

13. Click *OK*.

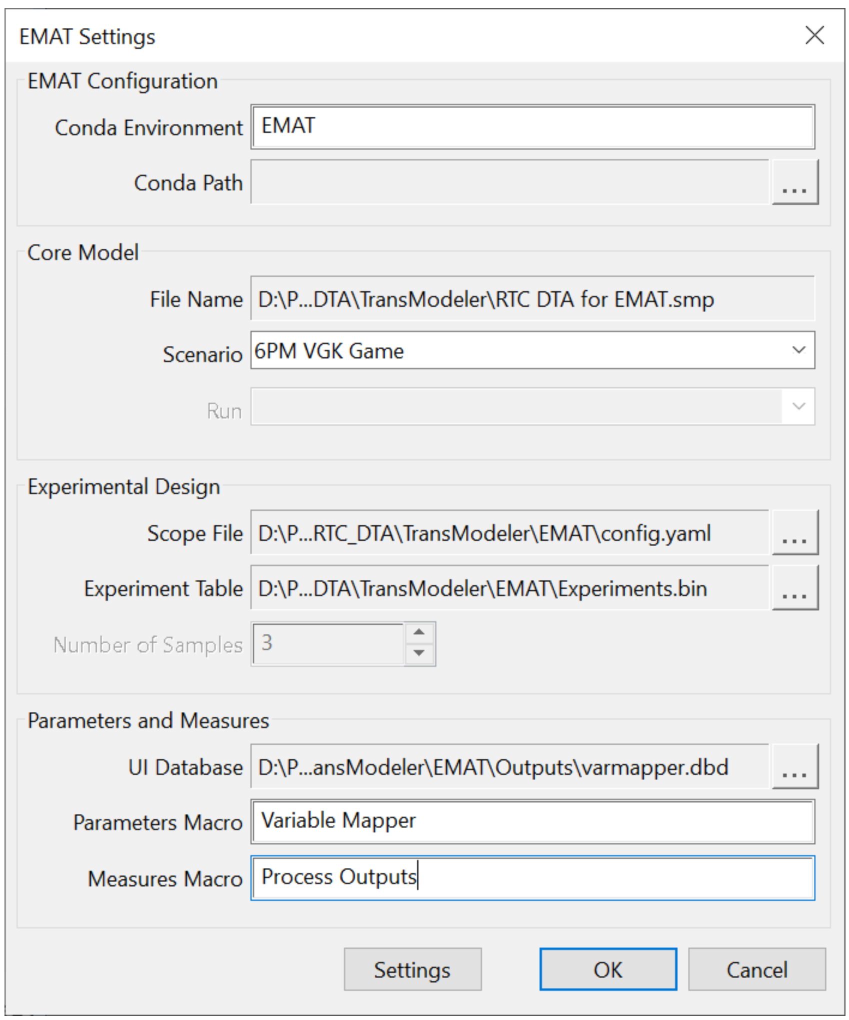


Figure 38. The EMAT Settings dialog box in TransCAD and TransModeler

To run EMA, perform the following steps:

1. Click the *Settings* button to enter the EMAT settings or to load them from previously saved settings.
2. Click *Create Experiment Table* to create the experiments according to the scope file and settings.
3. Click *Open Experiment Table* to review the experiments and factor values in a dataview.
4. Close the dataview before running the experiments, and then click *Run Experiments*. The core model will be run once for each record in the experiment table.
5. When the experiments have completed, click *Visualize Metrics* to create visualizations in TransCAD or TransModeler

You have the freedom to create visualizations in other software, for example in Excel, Jupyter, or other data science application. To do so, complete the following steps:

1. Click *Open Experiment Table* to open the experiments in a dataview. The output fields containing the performance measures will be populated for all experiments.
2. Choose *File > Save* and save the file as an Excel Spreadsheet or as a comma-separated value (CSV) file or other format that can be read into Jupyter or your data science application of choice.

The GISDK Interface for EMAT

TransCAD and TransModeler share a powerful platform that facilitates a wide range of data analysis and transportation modeling tasks. The GIS Developer's Kit, or GISDK, is the suite of tools that allow TransCAD and TransModeler to be customized to create valuable decision support systems for policy-makers. GISDK can also be used to develop customized visualization tools to translate model output to meaningful, actionable information. Such visualizations are especially helpful when working with the typically large volumes of spatial and non-spatial data generated by advanced travel demand models, including hybrid and activity-based models (ABM), or when analyzing the outputs of dynamic traffic assignments (DTA) or traffic simulations performed in TransDNA or TransModeler.

An EMAT GISDK class allows you to use GISDK to interface with TMIP-EMAT so that your travel model or DTA may serve as the core model with which EMAT performs its experiments. Specifically, you can use the EMAT class to design experiments, evaluate the constituent scenarios, organize and manage the outputs intuitively, and run scenario discovery analysis in order to measure and correlate the impacts of uncertainties in key model inputs.

An EMAT class was developed in GISDK to facilitate integration with EMAT. You can use the class to identify the scope file defining the experimental design, create an experiment table from the uncertainty variables and policy levers defined in the scope file, run the experiments, and visualize the performance measures. The GISDK class offers more power and flexibility to allow you to customize how EMAT experiments are prepared and executed, but knowledge of the class and its methods are not a requirement to use the EMAT application. The steps described

Constructor

EMAT(args)

Initializes a new instance of the Caliper EMAT class. This requires Anaconda, EMAT version 0.5.0, and the presence of the cairosvg package.

Properties

Name	Type	Contents
CondaPath	string	Optional. If conda.exe is not on the path, specify the full path to the conda executable or batch file
Environment	string	Name of the EMAT conda environment where EMAT is installed

Methods

AddScopeFile(OptionsArray option)

Adds the EMAT YAML scope file to the class

Option	Type	Description
FileName	String	Name of the YAML scope file

AddMetricsTable(string FileName)

Adds an existing experiment table to the class that can be used to visualize outputs.

AddMetrics(optarr Options)

Adds a specific output scatter plot to the visualizing dashboard

Options	Type	Description
TabName	String	Name of the output tab for this metric scatterplot
FieldX	string	Name of the output variable
FieldY	string	Name of the input variable

CreateExperiments(OptionsArray option)

Create a fixed format binary table with all the experiments based on the scope file

Option	Type	Description
DesignFileName	String	Name of the output experiment file name
ExtraArgs	optarr	Options array with options to create the experiments. See table below

ExtraArgs Options	Type	Description
addMetricFields	boolean	If true also adds the metrics field to the experiment table
n_samples_per_factor	int	Number of samples to draw for each input field

sampler	string	See TMIP-EMAT documentation. Name of the sampler ("lhs', 'ulhs', 'unu', 'ref')

FeatureScores()

Calculate the feature scores and add them to the visualizer dashboard

GetScopeInputs()

Returns the array of input variables in the scope file

GetScopeOutputs()

Returns the array of output variables in the scope file

LoadExperimentTable(string FileName)

Loads an existing experiment table. An experiment table is required to create dashboards of output metrics

PythonRunnerMode(string RunMode)

Optional. The default mode is 'Hidden'. Either 'Hidden', 'Normal', 'Minimized' or 'Maximized'

RunExperiments(OptionsArray options)

Runs a TransCAD planning model based on the input parameters and produces all the output metrics. If the planning model does not produce the matrix directly, a custom compiled macro needs to be provided that returns the required metrics.

Options	Type	Description
Model	optarr	Option array with planning model specification. See the table below
Measures	optarr	Optional. If model does not directly produce measures. Options to retrieve measures after a model run. See the table below
ExperimentTable	string	Name of the experiment table

Model Options	Type	Description
Name	String	Full path to the flowchart model file name
ScenarioName	String	Name of the scenario to use for the model runs
StepName	String	Optional. If provided only run this step in the model, otherwise the entire model runs

Measures Options	Type	Description
Function	String	Name of the function that produces the output metric
UI	String	Name of the compiled UI database containing the function that produces the output metrics

RunDnaExperiments(OptionsArray options)

Options	Type	Description
MapFileName	string	Name of the TransDna Map file with all the TransDNA scenario

		settings
ScenarioName	string	Name of the scenario for the TransDna model runs
ExperimentTable	string	Name of the experiment table
ParametersLookUp	optarr	Inputs for the lookup function to translate experiment table input fields to parameters in TransDna
Measures	optarr	Inputs for the function to calculate output measures from the TransDna run to be saved in the output fields in the experiment table

ParametersLookUp Options	Type	Description
Function	String	Name of the function that maps the parameters lookup
UI	String	Name of the compiled UI database containing the function that produces the output metrics

Measures Options	Type	Description
Function	String	Name of the function that produces the output metric
UI	String	Name of the compiled UI database containing the function that produces the output metrics

ThresholdFeatureScores(OptionsArray options)

Calculate threshold feature scores and add them to the visualizer dashboard

ViewExperimentTable()

Opens the experiment table into a dataview window

VisualizeMetrics(optarr Options)

Creates the output dashboard

Options	Type	Description
WindowTitle	string	Name of the TransCAD window
DashboardTitle	string	Name of the dashboard

Examples

```
// Example to visualize Output metrics
macro "GetFolder"
    // folder with configuration files
    return("e:\\projects\\EMAT_Tests\\MSAModel\\")
endmacro

macro "RunExperiments"
    {pInfo} = GetProgram()
    {disk, folder} = SplitPath(pInfo)
    folder = RunMacro("GetFolder")
    tut_folder = RunMacro("G30 Tutorial Folder")
    msaModel = tut_folder + "MSAModel\\MSA Feedback.model"
    configFile = folder + "msa_config.yaml"
```

```
    designFname = folder + "experiments.bin"
    obj = CreateObject("EMAT", {Environment: "emat"})
    obj.AddScopeFile({FileName: configFile})
    expTable = obj.CreateExperiments({
        DesignFileName: designFname,
        Args: {addMetricFields: True, n_samples_per_factor: 10, sampler: "lhs"}
    })
    obj.ViewExperimentTable()
    // ShowArray({obj.EmatData.ScopeInputFields, obj.EmatData.ScopeOutputFields})
    obj.RunExperiments({
        Model: {Name: msaModel, ScenarioName: "BaseGlobalBPR", Macro: null},
        ExperimentTable: expTable
        // , Measures: {Function: "outputMacroName", UI: "xxxx.db"}
    })
    // calculate feature scores and create a heat chart
    featureScoreImage = obj.FeatureScores()
    inputFields = obj.GetScopeInputs()
    outputFields = obj.GetScopeOutputs()
    // Create a dashboard
    obj.AddMetricsTable( expTable )
    for outField in outputFields do
        for input in inputFields do
            obj.AddMetrics({TabName: outField, FieldX: input, FieldY: outField})
        end
    end
    obj.VisualizeMetrics()
endmacro

macro "VisualizeOnly"
    on error do
        ShowMessage(GetLastError())
        return()
    end
    folder = RunMacro("GetFolder")
    tut_folder = RunMacro("G30 Tutorial Folder")
    msaModel = tut_folder + "MSAModel\\MSA Feedback.model"
    configFile = folder + "msa_config.yaml"
    designFname = folder + "experiments.bin"
    obj = CreateObject("EMAT", {Environment: "emat"})
    obj.AddScopeFile({FileName: configFile})
    obj.LoadExperimentTable(designFname)
    inputFields = obj.GetScopeInputs()
    outputFields = obj.GetScopeOutputs()
    FeatureScores = obj.FeatureScores()
    thresholdFigAlpha = obj.ThresholdFeatureScores({Measure: "GlobalAlpha"})
    inputFields = obj.GetScopeInputs()
    outputFields = obj.GetScopeOutputs()
    expTable = designFname
    // Create a dashboard
    obj.AddMetricsTable( expTable )
    for outField in outputFields do
        for input in inputFields do
            obj.AddMetrics({TabName: outField, FieldX: input, FieldY: outField})
        end
    end
    obj.VisualizeMetrics()
endmacro
```

Experiences Integrating EMAT with the Core Models

It is worth noting for the benefit of analysts attempting to perform analyses with the application in the future that a variety of technical difficulties may be encountered. Because the EMAT application relies on various layers of software communicating with one another via their own APIs, it can be difficult to ascertain the origins of an error when one arises that interferes with successful experiment execution or performance measure visualization. In the experiences of the project team developing the API in TransCAD and TransModeler that interfaces with EMAT and the team designing and running the experiments, problems routinely arose creating EMAT environments in Anaconda that worked consistently across all computers on which experiments were run. Occasionally, experiments that ran successfully on one computer would fail on another.

The most common source of the errors encountered stemmed from inconsistencies in the libraries that were successfully installed in the EMAT Anaconda environments. Toward the end of the project, it became impossible to calculate threshold feature score visualizations from EMAT on some computers but not on others. After an investigation of the error messages, it was determined that an incompatibility had arisen between the CairoSVG library on which EMAT relies to create the threshold feature score image. Investigation also revealed that the same error would occur when requesting the threshold feature score visualization via the Jupyter notebook recommended for visualization on the EMAT website. That error message is shown in Figure 40.

~a0r24.err

```
C:\Users\modeler\Anaconda3\envs\EMAT\lib\site-packages\sklearn\ensemble\_forest.py:396: FutureWarning:

Criterion 'mse' was deprecated in v1.0 and will be removed in version 1.2. Use `criterion='squared_error'` which is equivalent.

Traceback (most recent call last):
  File "<string>", line 1, in <module>
  File "C:\Program Files/TransModeler Dev/EMAT\EmatSupportFunctions.py", line 133, in calculate_threshold_feature_score
    cairosvg.svg2png(bytestring=outstr, write_to=outFigureName)
  File "C:\Users\modeler\Anaconda3\envs\EMAT\lib\site-packages\cairosvg\__init__.py", line 55, in svg2png
    return surface.PNGSurface.convert(
  File "C:\Users\modeler\Anaconda3\envs\EMAT\lib\site-packages\cairosvg\surface.py", line 131, in convert
    instance = cls(
  File "C:\Users\modeler\Anaconda3\envs\EMAT\lib\site-packages\cairosvg\surface.py", line 207, in __init__
    raise ValueError('The SVG size is undefined')
ValueError: The SVG size is undefined
```

Figure 39. Error message pointing to incompatibility between EMAT and a third-party library

The project team's experiences working with EMAT also revealed that casual familiarity with Python was not sufficient to make the EMAT experiments successful. Rather, proficiency with Python and Anaconda were necessary to uncover the root causes of errors such as the one depicted in Figure 19.

In summary, the RTC should anticipate that technical support installing and deploying EMAT may be required at times in its routine use. It may be challenging, however, to obtain technical support as and when help is needed given the open-source, volunteer-dependent nature of TMIP-EMAT. Hence, hiring staff that is conversant with Python or training staff to a reasonable degree of proficiency may be a step worth the RTC's consideration.

APPENDIX B: STAKEHOLDER INVOLVEMENT

A stakeholder involvement meeting was held on December 9, 2020 to elicit input in the EMA design. Specifically, the project team endeavored to learn which variables of uncertainty and which performance measures were most resonant with the stakeholders in the region's transportation system.

Stakeholders were invited from the following agencies:

- RTC
- Clark County
- Clark County Department of Aviation
- NDOT
- FHWA
- City of Henderson
- City of North Las Vegas
- Nellis Air Force Base (AFB)

Participants in the meeting also included consulting partners of RTC and members of the project team.

During the meeting, the concept of EMA was described, and attendees were asked to help the project team:

1. Better understand which aspects of risk and uncertainty impede decision-making.
2. Identify which policy levers or operational strategies are of interest in the context of said risks and uncertainties.
3. Identify which performance measures may help planners navigate risk and uncertainty in the decision-making process.

Variables of Risk and Uncertainty

The following ideas were raised as potential sources of risk and uncertainty in the region:

- Urban growth boundary (e.g., related to land use)
- Increase in freight trips/deliveries due to more e-commerce
- Transit-oriented development and its impacts on traffic and land use
- Special land uses
  - Amazon warehouses
- Forecast population (demographic/socioeconomic) growth
  - For example, Nellis AFB, around which growth estimates vary widely

  - Commuting patterns
  - Increased work from home
  - Telecommuting
- Connected and autonomous vehicles (including varying penetration, capacity impacts)
  - Self-driving shuttles
  - CAV operations on freeways
- Visitor traffic
  - Policies enticing visitors to the region
  - Airport trips
  - Mode choices and movements of tourists
  - High-speed rail (impacts on I15 road traffic and external trips; model currently uses most conservative estimates to reduce auto trips b/t CA and Las Vegas, easy to do high, medium, low scenarios; follow-up questions about hotel occupancy)
  - It was noted that the RTC travel model differentiates between single-day and multi-day visitors, but high-speed rail projections do not)
- Funding/costs
- Flying taxis

Policy Levers and Operational Strategies

The policy levers and operational strategies identified by the stakeholder group were:

- Road diets
  - Parkways/reconfigured streets to meet community goals
- Pedestrians
  - Scramble crosswalks, extended sidewalks
  - Longer crossing times (at signals)
- Transit service improvements
  - Transit signal priority (e.g., in the microscopic DTA model)
  - Improved shuttles for special events (sports, cultural, etc.) (perhaps only consider routes already in the model and their service characteristics, frequency of buses, etc.)
  - Bus lanes
- Traffic operational strategies on events
  - Christmas Eve
  - New Year's Eve
  - Arterial closures (e.g., to support foot traffic to T-Mobile Arena)
- Microsimulation DTA model operational improvements
  - Ramp metering
  - Signal timing
  - Dynamic message signs (DMS)
- Parking policies and pricing
- Complete streets concepts
- Bicycles
  - Buffered/protected bike lanes/paths
  - Regional Bicycle and Pedestrian Plan high-comfort network
- Other potential applications: special events such as Las Vegas Motor Speedway, races, and other speedway events such as the Electric Daisy Carnival

Performance Measures

The performance measures suggested and discussed included:

- Travel times
  - By mode
- Traffic delay due to incidents/crashes
  - Consider crash history data
- VMT (reduction)
  - Complete streets
  - Study existing plans for specific corridors
- Map-21 performance measures
  - Mode share (SOV, HOV, transit vs. drive)
- Equity in performance measures
  - e.g., travel times for higher- and lower-income TAZs
- Accessibility
- Cost-effectiveness of implemented transportation interventions at different time horizons
- Safety
  - Impacts of traffic concentration on safety performance
  - Crash frequency and crash rate
- Consider performance measures already in use in prior models/studies