# Simulation-Based Comparison of Sequential Optimization Methods for Personalized taVNS Parameter Selection

**Johnathan Ding**[1]

*Received August 17, 2026*
*Accepted October 13, 2026*
*Electronic access November 15, 2026*

Transcutaneous auricular vagal nerve stimulation (taVNS) is a cutting-edge form of neuromodulation that could provide more equitable access to vagal nerve stimulation for lower-income households or people at high risk of surgical complications. Optimizing taVNS parameters, such as amplitude and frequency, is difficult due to each person's unique physiology, however. Applying machine learning to tailor VNS parameters to each unique subject could improve this issue. In this fully synthetic experiment, Bayesian optimization algorithms Bayesian Optimization Gaussian Process, Tree-structured Parzen Estimator, Sequential Model-Based Algorithm Configuration, and a random search control were performed against simulated patients to determine the most effective machine learning optimizer for taVNS in the simulated experiment. A Friedman and paired permutation test was conducted to judge whether there was a statistical difference in effectiveness between the 4 optimizers. Only the Friedman test indicated a statistical difference ($p = .049$); the paired permutation test failed to find a statistical difference ($p = .137$). A post-hoc test showed no pair of optimizers had statistically significant differences in effectiveness after Holm correction. This simulation was purely synthetic, with the patient response being motivated by taVNS experiments, but not based empirically on them. Running a Bayesian optimizer-taVNS simulation on empirical data, or comparing Bayesian optimizers of taVNS in real life, is needed to see whether this simulation's results generalize to real life. In the future, Bayesian optimizers and/or other forms of machine learning should be explored with taVNS to maximize taVNS effectiveness for each unique individual.

**Keywords:** Vagal Nerve Stimulation, Bayesian Optimization, Neuromodulation, Neurology, Tree-structured Parzen Estimator, Sequential Model-based Algorithm Configuration.

## Introduction

Vagal Nerve Stimulation (VNS) is a type of neuromodulation of the vagus nerve. It stimulates the Locus Coeruleus LC and Nucleus Tractus Solitarius through the brainstem[1]. VNS treatment has been explored for various diseases, including depression, traumatic brain damage, Alzheimer's disease, and epilepsy[2]. VNS has ameliorative effects on brain diseases, one possible pathway being through neurotransmitters. The LC is responsible for norepinephrine (NE) production throughout most of the brain and shows upregulated cell activation via the nucleus tractus solitarius during invasive VNS (iVNS). iVNS is invasive due to its requirement of an expensive surgery to implant the neuromodulating device in the neck. This release of NE can improve symptoms in diseases by encouraging brain plasticity and decreasing neuroinflammation. Via stimulation of the medial septum, the neurotransmitter acetylcholine may also be released in VNS, binding to nicotinic receptors known as $\alpha$7nAChR in astrocytes and another glial cell called microglia, possibly reducing neuroinflammation through this cholinergic anti-inflammatory pathway. VNS can also regulate other neurotransmitters such as dopamine, serotonin, and GABA. Modulation of these neurotransmitters can explain its positive effects for neurological conditions[1].

VNS once required invasive surgery to implant the neuromodulation device onto the vagus nerve, thus being considered iVNS. Due to the costs and risks of the VNS method, noninvasive forms of VNS, one being Transcutaneous Auricular Vagal Nerve Stimulation (taVNS), are being investigated for use in diseases that iVNS treats, such as epilepsy, and other diseases, such as AD[3]. taVNS has been shown to act on common pathways in the NTS with Invasive VNS (iVNS) and evoke similar cerebral activation patterns as iVNS. Additionally, taVNS is known for its low cost and portability, with a 2026 systematic review reviewing taVNS application for major depressive disorder mentioning taVNS's similar effectiveness with iVNS in mental health treatment, and its advantages of noninvasiveness and low economic burden compared to iVNS[4]. taVNS is also under exploration for use in neurodegenerative

[1] *Acalanes High School, california, USA*

© The National High School Journal of Science 2026

conditions such as AD, as those patients are often older and have more comorbidities, increasing the risk of complications in iVNS implantation surgeries[5].

VNS effectiveness can be changed through altering VNS parameters such as intensity, frequency, pulse width, and duty cycle[6]. Intensity determines the amount of current that goes through the device, frequency determines the electrical pulses per second, pulse width determines the length of time electricity is passed through per pulse, and duty cycle is the percentage of time the VNS is on compared to the total stimulation time[7].

Finding the best parameters for taVNS is a troubling task for clinicians, however. A wide range of taVNS parameters, differing pain thresholds, and different physiological responses to the parameters based on factors such as age, sex, ear tissue anatomy, and time of day could possibly create difficulties in finding optimal taVNS parameters for brain activation in a patient[6,8,9]. There are also significant differences in vagus nerve fiber morphology from person to person, providing another possible reason for differing clinical reactions to VNS[10]. No consensus on optimal taVNS parameters exists[11], and a review article cites the need for extensive testing of different taVNS parameters in AD[12]. A way to adapt taVNS treatment to function optimally based on a subject's unique physiology would be helpful, and machine learning presents itself as a possible solution. Machine learning optimization could tune taVNS parameters, optimizing them to give the most effect for a patient. A study has demonstrated the feasibility of applying machine learning optimization in VNS, utilizing Bayesian Optimization Gaussian Process (BOGP) for invasive VNS in porcine subjects, showing Bayesian optimizers are well suited for supporting iVNS parameter optimization[13].

Few studies have looked at the application of machine learning algorithms to optimize taVNS parameters. A systematic literature review was done to confirm this potential literature gap.

## Search Strategy

A systematic review was conducted in the PubMed, IEEE, and Web of Science databases to identify research articles comparing the relative effectiveness of different forms of Bayesian optimization in taVNS.

The following search terms were entered into PubMed, IEEE, and Web of Science: ("transcutaneous auricular vagus nerve stimulation" OR "taVNS" OR "vagus nerve stimulation" OR "VNS") AND ("Bayesian optimization" OR "sequential model-based" OR "SMAC" OR "Tree-structured Parzen" OR "TPE" OR "hyperparameter optimization" OR "machine learning parameter tuning" OR "optimization algorithm parameter tuning").

This search query was confirmed to function in the 3 databases PubMed, IEEE, and Web of Science. The search was conducted in September 2026.

## Inclusion Criteria

Articles were included if they:

1. Mentioned taVNS and compared optimizers of all forms to the VNS or taVNS parameter tuning process.

Articles were excluded if they:

1. Were not in English
2. Did not reference the medical procedure of VNS
3. Did not compare the efficiency of parameter tuning optimizers.

9 articles were found in total: PubMed (n = 3), IEEE (n = 1), Web of Science (n = 5). 2 duplicate articles were then removed before screening. During screening, 3 articles were excluded based on title and abstract irrelevance. 4 articles were reviewed in full text, and no articles reviewed the application or comparison of Bayesian optimizers to taVNS procedures. Only 1 article was found to compare Bayesian Optimization Gaussian Process (BOGP) with the control parameter tuning optimizer, random search, in iVNS, providing adjacent evidence. This review highlights a literature gap on finding the comparative effectiveness of Bayesian optimizers in optimizing taVNS parameters.

## Heart Rate Variability and Pupillometry as VNS metrics

Heart rate variability (HRV) and pupillometry are noninvasive measures of the autonomic nervous system and brain arousal, both of which can be affected indirectly under taVNS stimulation. They have thus been used as indirect indicators of taVNS effectiveness, with a study using both metrics concurrently when experimenting on taVNS's effect on working memory[14], demonstrating the feasibility of using both metrics at once to measure taVNS effect. For this simulation, HRV and pupillometry are simply used as simulated response components.

HRV functions as a metric of autonomic nervous system function, which is made up of the sympathetic and parasympathetic nervous systems. This metric measures millisecond changes in time in heartbeat fluctuations. Because taVNS has an effect on the autonomic nervous system, studies have used HRV measurements to indirectly measure taVNS effectiveness. A 2014 study conducted on 48 participants found that taVNS significantly increased HRV, signaling a decrease in sympathetic nervous system activity[15]. However, a 2021 meta-analysis of 16 articles determining whether taVNS has an effect on vagally mediated HRV found stronger evidence


© The National High School Journal of Science 2026

for the null hypothesis[16]. HRV is not a conclusive indicator of taVNS effectiveness and shows wide metric variability between studies.

Pupillometry is a measure of neurological arousal and of Locus Coeruleus (LC) NE production, thus acting as an alternative but indirect proxy for measuring NE levels, as there is a lack of noninvasive devices that directly measure NE in human brains. An increase in LC neural firing through iVNS has been shown to increase pupil dilation in rats. A study found that taVNS induced significant pupil dilation in an experiment done among 24 healthy males[17]. In another study, iVNS was shown to increase pupil dilation in humans[18], with the pupillometry response in humans following an inverted U response based on VNS amplitude. This study also demonstrated the inverted-U pupillometry response closely aligned with a Gaussian or a sigmoid function distribution.

Similar to HRV, there are studies that give conflicting evidence of pupillometry as a reliable metric. A study stating pupillometry as likely the most reliable noninvasive marker of LC-NE activity found null results with pupillometry when subjects were treated with taVNS[19]. A meta-analysis of 18 articles determining if taVNS alters pupil dilation found null results with conventional taVNS protocols, and only found evidence for pupil dilation with pulsed taVNS[20]. A large 2026 experiment with 94 healthy participants found no increased pupil dilation effect by pulsed taVNS when the sham device mimicked the sensation of the real taVNS device[21].

Both HRV and pupillometry have been acknowledged as promising noninvasive indicators of taVNS effectiveness in studies, but demonstrate both positive and null results. They are under scrutiny for this demonstrably mixed effectiveness. They are noninvasive and relatively quick metrics that might theoretically suit the application of taVNS hyperparameter tuning, which would require multiple taVNS readings in a real-life given patient to adapt to their physiology. They aren't proven, direct indicators of taVNS's cognitive efficacy, but they'll still be used as simulated objective components for the sake of this experiment.

### Machine Learning Optimization Algorithm Types

Bayesian Optimization (BO), also termed sequential model-based optimization (SMBO), is a family of machine learning algorithms that are able to "learn" from previous data points. Simply, when modeling a real-life function such as a real person's pupillometry response to taVNS parameters, the algorithm creates a simplified function, or the surrogate model, and attempts to optimize or minimize a value with the finite amount of data points it is allowed to collect. A Bayesian optimizer consists of 2 parts: a statistical model that calculates the uncertainty of the values on a real-life function based on known data points, and an acquisition function, a function that decides the next point to sample in the real-life function, or objective function, to gather data or improve the modeled function. This acquisition function balances exploration vs exploitation. It chooses points on the real-life function, or objective function, that will likely maximize the surrogate function's value, or decrease the uncertainty of the objective function. In general, accuracy of the statistical model is more important than the acquisition function, as an inaccurate model of the objective function cannot be made up for with better decisions on prioritizing exploration vs exploitation via the acquisition function[22].

There are three main subtypes of SMBO that will be tested in this experiment: BOGP optimization, tree-structured parzen estimator (TPE) optimization, and sequential model-based algorithm configuration (SMAC) optimization. These Bayesian optimizers differ in the ways they construct their statistical model and also slightly differ in their acquisition functions, and can thus be stronger or weaker in different optimization scenarios.

### Bayesian Optimization Gaussian Process

In Bayesian Optimization Gaussian Process (BOGP), every point on the surrogate model has 2 values: the predicted value and its predicted variance/uncertainty. When calculating the statistical map, the values far from tested values will have high uncertainty, while values next to a tested value will have very low uncertainty. As the optimizer tests more numbers, the statistical model also fits a value that controls the scale through which this certainty will decrease given the point's distance from an established point. For example, if you have a very unpredictable curve, a point far out will probably have extremely high variance, while a slow-changing curve will not see as much variance in a point the same distance away. This allows for an adjustment in variance values for points on really sensitive objective functions, or less variable objective functions.

To calculate the value of each point and its uncertainty, BOGP predominantly uses a kernel known as Matern. This kernel differs from the other optimizers in that it is unable to handle categorical variables. As stated above, it attempts to assign a predicted value along with an uncertainty value to every point in the objective function through mathematics. Thus, variables like "blue" and "red" cannot be put into BOGP. As such, BOGP is restricted to quantitative optimization, unlike TPE and SMAC, which can handle categorical optimization.

BOGP's acquisition function, or the function selecting the next point on an objective function based on the statistical graph of the objective function, is usually a function called Expected Improvement. This function, when trying to find the target objective value, the best reachable value in the objective function, factors in the size of the gain that may occur, along with the probability that the gain will occur. This allows it to

balance trying to capitalize on, say, the local max in a small portion of the graph with experimenting with other areas on the objective function to find the global max[23].

BOGP has already been used as a way to optimize iVNS parameters in porcine subjects as previously mentioned, with the study stating it converges on optimal stimulation of the nerves with far fewer tests than would have been the alternate optimization method random search, reducing potential side effects and demonstrating the possibility of optimization algorithm implementation into VNS[13].

### Tree-Structured Parzen Estimator

Tree-Structured Parzen Estimator (TPE) is another form of Bayesian optimization. It shares the acquisition model Expected Improvement with BOGP, but differs in the statistical model it constructs.

Imagine you put 3 continuous variables into an objective function, and it gives an output variable, which you want to optimize. TPE separates the input parameters into a "good" function map and an "average" function map. Parameters are considered good if they're within the top n% of output values of the function, commonly used numbers being 15% or 25%, and considered mediocre if they are not. If the parameters are considered good, normal distributions with their means at each input variable are added to a "good" map line created for each variable. The good map line for each variable is denoted by L(x1), L(x2) and et cetera.

If the 3 variables returned a mediocre output value, not within the 15% or 25%, normal distributions with each mean at each input variable are put onto each respective "mediocre" map line that was created for each variable. This mediocre function is denoted by G(x1), G(x2), and et cetera, for each map line.

L(x1), L(x2), and any other variable's good map line are all multiplied together to create L(x), as well as all the mediocre G(x) map lines. This creates a statistical map of good parameter values and bad parameter values.

The fraction L(x) / G(x) is then used as the acquisition function for TPE to predict the next location to test. The Expected Improvement function from BOGP is similar to TPE's acquisition function, as maximizing the TPE ratio is equivalent to maximizing the expected improvement[24].

### Sequential Model-based Algorithm Configuration

Sequential Model-based Algorithm Configuration (SMAC) uses multiple decision makers, or trees, which have access to different subsets of trials to make its statistical model. SMAC first gets data from a few trials. Then, it assigns random subsets of those trials to each tree with replacement, there being usually 10 trees in total. For example, if 5 trials have been run, tree 1 may be trained with trial 1, 2, 2, 4, and 5. Tree 2 could be trained with trial 1, 1, 3, 5, 5. In each trial, a series of conditional inequalities are made to separate those trials into groups with the least variance between their output scores. For example, Tree 1 got output scores 90, 92, 92, 96, and 21 for its respective assigned trials. Let's say the first 4 trials had an amplitude of 2.0, but the 5th had an amplitude of 1.0. The tree would create a condition checking whether the amplitude is more than 1.5. If so, it'd assign the average of the 4 outputs. If not, it'd assign the average of the 2nd group, which is simply 21 because the 5th trial stands alone.

Each tree creates its own conditions through the trials it is assigned. Through this, the 10 trees can act similarly to BOGP, determining the predicted value and variance of each point on the objective function through a combined average decision on each point based on the output that each tree predicts. Once a new trial is conducted, random trials with replacement are assigned to every tree again, and the statistical map is recreated[25].

This explains SMAC's method of constructing a statistical model. For its acquisition function, it is able to utilize Expected Improvement, but usually utilizes Log Expected Improvement. Log expected improvement is preferred over Expected Improvement in SMAC because when testing a far-off unknown value, say a really low amplitude, many trees may default to their lowest inequality. This may make the variance extremely low, which could thus cause an underflow error when calculating the Z score of the expected improvement function, which is assigned to each point based on its expected magnitude of improvement and amount of uncertainty. Log expected error allows the Z score not to underflow to 0, but to remain as a very small number[26].

BOGP tends to have higher efficacy in low-parameter and continuous parameter environments. TPE tends to have higher effectiveness in terms of speed, but struggles to capture parameter interdependence, and requires many startup evaluations. SMAC is stronger with categorical variables than BOGP, but may be less effective in continuous parameter environments due to its use of piecewise approximations.

A simulation will be conducted to determine the Bayesian optimizer that responds most effectively to the response surface families it is put through.

## Methodology

### Study Design

This is a fully synthetic experiment. The simulation wasn't built directly modeling empirical clinical taVNS data. Instead, due to a lack of standardized taVNS procedures and consistent measurement metrics, published taVNS parameter ranges and clinical taVNS response patterns were used to simply motivate


© The National High School Journal of Science 2026

the search bounds and response surface family creation. Also, once again, for a reminder, HRV and pupillometry scores are only coded simulation objectives.

### Response-Surface Model

The patient response function takes in 3 parameters: Amplitude, Frequency, and Pulse-width.

Studies on rats have shown amplitude parameters may create an inverted U shape with regard to the effectiveness of VNS[27,28] VNS possibly demonstrates this shape of effectiveness drop-off with regard to frequency and pulse-width also, in a study done on healthy adults[8].

This shape isn't clinically confirmed, due to a lack of studies on taVNS distributions, but will be implemented into the patient response function, as the inverted U curve theory is still a recurring theme in VNS scientific literature.

$$H_{x,\mu,L,R} = \frac{1}{1+\left(\frac{x-\mu}{s_x}\right)^4} \tag{1}$$

$$s_x = L \quad \text{when } x < \mu, \quad s_x = R \quad \text{when } x \geq \mu.$$

$$A_a = 50H_{a;\mu_{\text{pup}},L_{\text{pup}},R_{\text{pup}}} + 50H_{a;\mu_{\text{HRV}},L_{\text{HRV}},R_{\text{HRV}}} \tag{2}$$

$$Q_{a,f,w} = H_{f;f_{r,a},L_f,R_f} \cdot H_{w;w_{r,a},L_w,R_w}$$

$$J^{\text{NF}}_{a,f,w} = A_a \cdot Q_{a,f,w} \qquad J^{\text{obs}}_{a,f,w} = J^{\text{NF}}_{a,f,w} + \varepsilon \tag{3}$$

This shows the functions describing the simulated patient response. In 1), the asymmetric hill function H takes in the input parameter x, the optimal parameter value $\mu$, and a left and right scale. When the input score is less than the optimum, the function uses the left scale.

When the input score is more than the optimum, the function uses the right scale. This allows for an asymmetric hill function response. At 2), function A represents the amplitude, which is made up of the equally weighted sum of pupillometry and HRV. The amplitude score for pupillometry and HRV is calculated using the asymmetric Hill function effectiveness drop-off for pupillometry and HRV, and adding them together.

$Q(a,f,w)$ is the max value of the product of the K number of simulated peaks, representing the multiplied frequency and pulse-width score. f represents the frequency, and w represents the pulse width. At 3), the A and Q functions are multiplied together to create the noise-free function J. $J^{\text{obs}}$ represents the observed values by the optimizers, and epsilon represents the random noise score added to the noise-free function J to make $J^{\text{obs}}$.

As demonstrated, the pupillometry and HRV metrics each have a weight of 0.5. The figure demonstrates the whole method by which the calculated score, and therefore simple regret, is calculated. Using the max possible score along with simple regret, this allows for the creation of normalized regret, which allows for better comparison of different optimizers because simple regret is scaled to the max possible score the optimizer could actually reach.

There is also a unique surface response family, known as the unique parameter interaction family. The 2 optimal frequency and pulse-width locations change location depending on the amplitude parameter in that family.

Their movement is ensured in this way not to go out of the search boundaries: If the optimal frequency is greater than or equal to 25 Hz, then the optimal frequency decreases by the raw guessed amplitude. If the optimal frequency is less than 25 Hz, then the optimal frequency increases by the raw guessed amplitude.

For pulse width, if the optimal pulse width is greater than or equal to 350 $\mu$s, the optimal width decreases by the raw guessed amplitude multiplied by 20. If the optimal pulse width is less than 350 $\mu$s, then the optimal pulse width increases by the raw guessed amplitude multiplied by 20.

### Simulated Profile Generation

**Table 1** This table describes the scales of the asymmetric hill functions of the different optimizers. U stands for uniform, a randomly chosen value between the 2 numbers based on the patient and machine start seed.

| Family | Optima | Amplitude Rise/Fall scales | Frequency rise/fall scales | Pulse-width rise/fall scales. |
|---|---|---|---|---|
| Quick_rise_slow_fall (Baseline) | 2 | Pup: U(0.25,0.35) /U(0.55,0.75); HRV: U(0.2, 0.3) / U(0.45,0.65) | U(1,2)/U(5,6) Hz | U(20,50) /U(125, 175) $\mu$ seconds |
| Symmetric | 2 | Pup: 0.25; HRV: 0.25 mA | 1.5 / 1.5 Hz | 75 / 75 $\mu$ seconds |
| slow_rise_quick_fall | 2 | Pup: U(0.55, 0.75) / U(0.25,0.35); HRV: U(0.45, 0.65) / U(0.2,0.3) | U(5,6) / U(1,2) Hz | U(125, 175) / U(20,50) |
| 3 Optima | 3 | Same as quick_rise_slow_fall | Same as quick_rise_slow_fall | Same as quick_rise_slow_fall |
| broad_optima | 2 | Quick_rise_slow_fall amplitude falloffs, multiplied by 2 | Quick_rise_slow_fall frequency falloffs, multiplied by 2 | Quick_rise_slow_fall pulse-width falloffs, multiplied by 2 |
| sharp_optima | 2 | Quick_rise_slow_fall amplitude falloffs, multiplied by 0.4 | Quick_rise_slow_fall frequency falloffs, multiplied by 0.4 | Quick_rise_slow_fall pulse-width falloffs, multiplied by 0.4 |
| heteroscedastic, heavy-tailed, interaction | 2 | Same as quick_rise_slow_fall | Same as quick_rise_slow_fall | Same as quick_rise_slow_fall |

**Table 2** Table describing epsilon from Section "Response Surface Model" Demonstrates how the random noise is added depending on the optimizer family.

| Family | Noise Distribution |
|---|---|
| Heteroscedastic | As the amplitude inputted as a parameter increases, the standard deviation of the noise added to the noise-free function increases. The standard deviation matches the amplitude number directly, with an input of 5.0 mA setting a 5-point standard deviation. |
| Heavy-tailed noise | The noise added to the score follows a Student's t distribution with 3 degrees of freedom, multiplied by 2. |
| Other Families | A normal distribution with a mean of 0 and a standard deviation of 2 is used to create the random score. That score is then added on top of the noise-free function like the others as epsilon in the graph from Figure 1. |

### Random Noise Score

It should be noted in advance that truncation rules were not added to this simulation. Thus, adding the noise distributions present in Table 2 on top of the noise-free function can possibly produce scores outside of the 0 to 100 range. This will be mentioned in the discussion.

### Optimizer Configurations

A computational simulation based on clinical taVNS parameter boundaries and reported taVNS effectiveness function shapes was created for this experiment. Here are the models and variables for creating this simulation.

Libraries used:

- Python: 3.13
- NumPy: 2.1.3
- SciPy: 1.16.3
- Scikit-learn: 1.6.1

Optimizer search bounds: Amplitude: [0.5, 5.0] mA; Frequency: [1.0, 50.0] Hz; Pulse width [50, 600] $\mu$seconds; These search boundaries are modeled after an international review article on taVNS, which included taVNS study parameter data, and an experiment that tested many frequency and pulse-width combinations[9,29].

Trials/queries per optimizer: 40; Permutations per inferential test: 20000; Bootstrap replicates per confidence interval: 10000; Regret tolerance: 0.05; Analysis random seed: 10001

### Optimizer Presets

**Table 3** Settings, such as initializations, priors, and acquisition functions to configure each Bayesian optimizer.

| Optimizer | Config |
|---|---|
| BOGP | Bayesian-optimization version 3.3.0; Expected Improvement with xi value of 0.01; Initial random sample of 4 points; Iterations = 36; Random state = profile seed; Matérn kernel with length scale 1.0, nu = 2.5; GP alpha = 1 * $10^{-6}$; n_restarts_optimizer = 5; Optimizer = fmin_l_bfgs_b; Duplicate points disallowed. |
| TPE | Hyperopt version 0.3.0; Uniform distributions across all parameters; Prior weight 1.0; Startup jobs = 20; Expected Improvement with candidates = 24; Gamma = 0.25 (Defines "Good/Bad" division as has been mentioned); np.random.default_rng(profile_seed); return_argmin=True; max_queue_len = 1; |
| SMAC | SMAC version 2.4.1; Config Space ver. 1.2.2; Iterations = num_simulation_trials times per patient; Random forest model with 10 trees; Feature ratio of 1.0; Minimum split of 2; Minimum leaf of 1; Maximum depth of 1048576; Bootstrapping enabled; (The default) Log Expected Improvement with xi value of 0; Sobol initial design with n_configs = None; 10 configurations per hyperparameter, but max ratio of 0.25; Random design probability of 0.2; Intensifier: max_config_calls = 3, max_incumbents = 10; Config selector: Retrain_after = 8, retries = 16; Deterministic = false; 1 worker; |
| Random Search | Independent Uniform selections per parameter per trial, with replacement. Random number generation = np.random.default_rng(profile_seed) Runs num_simulation_trials times per patient. |

In Table 3, these settings are usually the default settings that these optimizers are set to. No hyperparameter tuning was performed. Settings not explicitly stated here use the default settings from packages.

### Other Experimental Variables

In this simulation, the statistical tests will importantly use seed block pairing between the optimizers' results. Each optimizer will be run through 9 families with 40 profiles per family. Directly treating the 360 profiles from an optimizer as independent and therefore conducting paired permutation tests or Friedman tests with each 360-profile set would be erroneous


© The National High School Journal of Science 2026

**Table 4** Other variables that were set in order to make the simulation function.

| Variable | Quantity | Purpose |
|---|---|---|
| patient_and _machine_start _seed | 10000 | Used as the starting seed for all trials. |
| 1st optimum frequency & pulse-width | 10 + U(-2,2) Hz, 500 + U(-25,25) $\mu$seconds | Generates a patient's first optimal frequency and pulse-width combination. |
| 2nd optimum frequency & pulse-width | 25 + U(-2,2) Hz, 100 + U(-25,25) $\mu$seconds | Generates a patient's second optimal frequency and pulse-width combination. |
| 3rd optimum frequency & pulse-width | U(5,50) Hz, U(150,450) $\mu$seconds | Generates a 3rd combination for the patient, only if from the 3 optima family. |
| Hill exponent | 4.0 | Dictates the steepness of the hill function falloffs, with lower numbers being less steep. |
| Ideal pupillometry amplitude | Normal(mean = 4.4, standard deviation = 0.15) mA. | Using a normal distribution, chooses a random amplitude number to be the most effective. |
| Ideal heart rate variability amplitude | Normal(mean = 4.0, standard deviation = 0.15) mA | Chooses an amplitude number that would make HRV metric respond most effectively for a given patient. |

due to the shared profile-generation seeds between each of the 40 profiles. Doing so would artificially lower the p-value. Thus, seed block pairing was done.

Essentially, the test function starts with a seed of 10000. Then, for all the patient profiles in the 9 response surface families that share that first seed, an optimizer's mean normalized regret score across the nine patient profiles with that first seed is calculated. We do this for every seed.

Then, we compare the four optimizers across the 40 seed blocks using a Friedman omnibus test and a paired permutation test; pairwise comparisons between the optimizers are then applied with seed-level sign-flip tests and Holm correction. The sample size is also thus considered 40 seed blocks, and not 360 independent profiles. All profiles related by the same patient and machine start seed are paired.

The simulation code can be accessed here: `https://github.com/BurningAesthetics/Simulation-Based-Comparison-of-Sequential-Optimization-Methods-for-taVNS-Parameter-Selection`

## Results

The Bayesian Optimizers demonstrated differing responses based on response families. In Tables 5 and 6, the tables demonstrate the effectiveness, standard deviation, and computation time of the quick rise slow fall (baseline), symmetric, slow rise quick fall, 3 optima, and broad optima families.

Interestingly, the broad optima showed lower scores than the mean/median of the other families, likely because broader, larger responses of the family made parameter selection easier. Conversely, in Table 6, the sharp peaks family had relatively consistently high normalized regret scores, highlighting a difficulty in searching for the optimum values. This difficulty was also reflected in the symmetric family.

**Table 5** Statistics of the optimizers' performance across the simulated response-surface families. The table shows mean and median normalized regret, standard deviation, mean simple regret, probability of regret within the prespecified tolerance, and mean computational time. Lower regret values mean better optimizer performance, and the probability of regret utilizes the normalized regret. Interestingly, TPE had a much lower standard deviation and mean normalized regret when performing in the broad optima family than the other families.

| Family | Optimization Method | Mean Norm. Regret (%) | Median Norm. Regret (%) | SD Norm. Regret (%) | Mean Simple Regret | Prob. Regret $\leq$ Tol. | Mean Comp. Time (s) |
|---|---|---|---|---|---|---|---|
| Quick_rise_slow _fall (Baseline) | BOGP | 45.88 | 37.12 | 37.34 | 42.96 | 0.150 | 10.88 |
| | TPE | 44.42 | 41.77 | 29.52 | 42.04 | 0.100 | 0.14 |
| | SMAC | 58.26 | 52.92 | 36.91 | 53.63 | 0.000 | 7.40 |
| | Random Search | 64.67 | 67.18 | 29.82 | 60.67 | 0.025 | 0.01 |
| Symmetric | BOGP | 76.32 | 99.92 | 35.47 | 60.79 | 0.075 | 6.18 |
| | TPE | 82.85 | 100 | 29.19 | 64.01 | 0.000 | 0.16 |
| | SMAC | 78.34 | 98.06 | 31.04 | 60.81 | 0.050 | 8.09 |
| | Random Search | 80.91 | 90.93 | 23.53 | 63.06 | 0.000 | 0.01 |
| Slow_rise_quick _fall | BOGP | 57.58 | 71.43 | 38.93 | 55.82 | 0.250 | 7.31 |
| | TPE | 59.74 | 59.55 | 29.35 | 57.69 | 0.025 | 0.17 |
| | SMAC | 65.78 | 84.41 | 35.37 | 62.38 | 0.025 | 8.20 |
| | Random Search | 56.58 | 53.24 | 28.15 | 53.18 | 0.025 | 0.01 |
| 3_optima | BOGP | 41.13 | 29.75 | 35.26 | 37.92 | 0.275 | 14.73 |
| | TPE | 41.41 | 41.72 | 25.00 | 37.65 | 0.075 | 0.17 |
| | SMAC | 49.50 | 45.04 | 34.23 | 45.41 | 0.075 | 7.12 |
| | Random Search | 50.90 | 54.63 | 27.77 | 46.82 | 0.000 | 0.01 |
| broad_optima | BOGP | 24.98 | 15.95 | 23.73 | 24.66 | 0.250 | 22.68 |
| | TPE | 9.04 | 7.46 | 8.29 | 8.94 | 0.375 | 0.17 |
| | SMAC | 19.39 | 11.19 | 20.73 | 18.9 | 0.300 | 6.82 |
| | Random Search | 22.13 | 17.41 | 17.76 | 21.83 | 0.200 | 0.01 |

Through the data shown in Table 7, the best performances of the different optimization methods are demonstrated. The quick rise slow fall family, considered the baseline family, was best optimized by TPE optimization, with the mean normalized regret being 44.42.

The symmetric family was best optimized by Bayesian optimization with a mean normalized regret of 76.32. Slow rise quick fall family was best optimized by random search with a mean normalized regret score of 56.58. The 3 optima family was best optimized by Bayesian optimization with a mean

**Table 6** Continued summary including the optimizer methods and the sharp optima, heteroscedastic noise, heavy-tailed noise, and unique parameter interaction families. Between the heteroscedastic and heavy-tailed noise families, one optimizer interestingly did not come out as most effective regarding handling noise. TPE demonstrated a lower median normalized regret with the heteroscedastic noise, while BOGP did so with the heavy-tailed noise family.

| Family | Optimization Method | Mean Norm. Regret (%) | Median Norm. Regret (%) | SD Norm. Regret (%) | Mean Simple Regret | Prob. Regret $\leq$ Tol. | Mean Comp. Time (s) |
|---|---|---|---|---|---|---|---|
| sharp_optima | BOGP | 87.47 | 100 | 23.09 | 58.11 | 0 | 5.63 |
| | TPE | 96.98 | 100 | 8.71 | 63.36 | 0 | 0.17 |
| | SMAC | 91.08 | 100 | 18.53 | 59.17 | 0 | 8.96 |
| | Random Search | 93.05 | 100 | 16.62 | 60.99 | 0 | 0.01 |
| heteroscedastic_noise | BOGP | 47.12 | 42.51 | 37.27 | 44.36 | 0.225 | 11.20 |
| | TPE | 48.09 | 35.89 | 33.95 | 44.40 | 0.100 | 0.19 |
| | SMAC | 59.10 | 55.24 | 34.85 | 54.75 | 0.050 | 8.53 |
| | Random Search | 65.33 | 67.18 | 30.48 | 61.32 | 0.025 | 0.01 |
| heavy_tailed_noise | BOGP | 47.28 | 44.59 | 36.81 | 44.51 | 0.175 | 15.72 |
| | TPE | 56.40 | 62.24 | 31.40 | 52.56 | 0.025 | 0.19 |
| | SMAC | 61.75 | 62.24 | 36.04 | 57.18 | 0.075 | 10.28 |
| | Random Search | 65.04 | 67.18 | 30.29 | 61.03 | 0.025 | 0.01 |
| unique_parameter_interaction | BOGP | 53.99 | 61.50 | 31.52 | 50.31 | 0.100 | 11.92 |
| | TPE | 31.76 | 27.92 | 24.13 | 29.94 | 0.125 | 0.16 |
| | SMAC | 61.06 | 66.50 | 33.00 | 57.07 | 0.075 | 8.41 |
| | Random Search | 53.85 | 60.14 | 30.50 | 49.97 | 0.000 | 0.01 |

**Table 7** A table describing the mean normalized regret score of each family with their Bayesian optimizer.

| Iteration Budget | Surface Family | Bayesian Optimization | TPE Optimization | SMAC Optimization | Random Search |
|---|---|---|---|---|---|
| 40 | 3optima | 41.13 | 41.41 | 49.50 | 50.90 |
| 40 | broad_optima | 24.98 | 9.04 | 19.39 | 22.13 |
| 40 | heavy_tailed_noise | 47.28 | 56.40 | 61.75 | 65.04 |
| 40 | heteroscedastic_noise | 47.12 | 48.09 | 59.10 | 65.33 |
| 40 | quick_rise_slow_fall | 45.88 | 44.42 | 58.26 | 64.67 |
| 40 | sharp_optima | 87.47 | 96.98 | 91.08 | 93.05 |
| 40 | slow_rise_quick_fall | 57.58 | 59.74 | 65.78 | 56.58 |
| 40 | symmetric | 76.32 | 82.85 | 78.34 | 80.91 |
| 40 | unique_parameter_interaction | 53.99 | 31.76 | 61.06 | 53.85 |

normalized score of 41.13. With a mean normalized score of 9.04, TPE best optimized the broad optima family.

Bayesian optimization best optimized sharp optima with a mean normalized regret score of 87.47. Heteroscedastic noise was technically best optimized by BOGP optimization with a mean normalized regret score of 47.12, although TPE had a lower median normalized regret score of 35.89 compared to BOGP's median normalized regret score of 42.51.

Heavy-tailed noise was optimized best by Bayesian optimization, with a mean normalized regret score of 47.28. Finally, unique parameter interaction was best optimized by TPE optimization with a normalized regret score of 31.76.

## Overall Descriptive Performance Across Simulated Families

**Table 8** Optimizer performance statistics after 40 evaluations per profile across 9 families with 40 profiles per family, 360 profiles per optimizer. Demonstrates the mean normalized regrets of each of the optimizers, the 95% confidence interval of where their mean normalized regrets lie, probability of reaching a 5 or less normalized regret, how many families won, and mean seconds of computation time per synthetic patient.

| Method | Mean Regret (%) | 95% Seed CI Low (%) | 95% Seed CI High (%) | Median Norm. Regret (%) | Success Rate ($\leq$ 5%) | Mean Profile Rank | Families Won | Mean Seconds |
|---|---|---|---|---|---|---|---|---|
| Bayesian Optimization | 53.53 | 47.15 | 59.87 | 60.48 | 0.17 | 2.42 | 5 | 12.6 |
| TPE Optimization | 52.3 | 48.02 | 56.61 | 45.69 | 0.09 | 2.29 | 3 | 0.17 |
| SMAC Optimization | 60.47 | 53.99 | 66.96 | 65 | 0.07 | 2.64 | 0 | 8.26 |
| Random Search | 61.38 | 55.97 | 66.62 | 64.35 | 0.03 | 2.65 | 1 | 0.01 |

In Table 8, 95% "seed block" confidence intervals are calculated. For context, the 95% seed-block CI is the confidence interval of the mean normalized regret of each of the 40 profile seeds for that optimizer.

Remember, each profile seed consists of an average normalized regret of the 9 families that all use the same seed.

## Global Paired Omnibus Tests

**Table 9** Only the Friedman test indicates a statistical difference in the effectiveness of optimization methods. The paired permutation p-value fails to find a statistical difference. Kendall's W value of 0.066 indicates the difference in mean normalized regret is small. Thus, this omnibus test gives mixed results for a significant difference between the optimizers.

| N Paired Seeds | N Families | Weighting | Endpoint | Friedman Stat. | Friedman P | Kendall's W | Perm. Range Stat. | Perm. P |
|---|---|---|---|---|---|---|---|---|
| 40 | 9 | Equal | Mean norm. noise-free regret | 7.86 | 0.049 | 0.066 | 0.091 | 0.137 |

## Global Paired Post-hoc tests with Holm correction

Throughout the surface-response families, it's been noticed that TPE tends to have a lower standard deviation than BOGP, as shown in Tables 5 and 6.

In fact, in all 9 optimizer families, BOGP had a higher normalized regret variability than random search. In specific instances, TPE performed surprisingly well, not only performing well in just the broad optima family as shown in Table 5.

In Table 8, TPE's average computation time per simulated patient profile is a surprising 0.17 seconds, while BOGP's average computation time per simulated patient is 12.6 seconds.


© The National High School Journal of Science 2026

**Table 10** This table shows the results of post hoc tests on the optimizer family pairs, with Holm correction applied. The mean normalized regret difference is shown, as well as the 95% seed-block bootstrap confidence intervals. Paired effect size in standard deviations, proportion of seed blocks where method A has lower regret, paired sign flip values, and Holm-adjusted p-values are also shown.

| Method A | Method B | N Seeds | N Fam | Diff (A-B) (%) | CI Low | CI High | Effect $d_z$ | Prop A Lower | Sign-Flip P | Holm P |
|---|---|---|---|---|---|---|---|---|---|---|
| Bayesian | TPE | 40 | 9 | 1.23 | -5.33 | 7.51 | 0.06 | 0.48 | 0.71 | 1 |
| Bayesian | SMAC | 40 | 9 | -6.95 | -16.72 | 3.08 | -0.21 | 0.6 | 0.19 | 0.56 |
| Bayesian | Random | 40 | 9 | -7.85 | -16.62 | 1.27 | -0.27 | 0.6 | 0.09 | 0.36 |
| TPE | SMAC | 40 | 9 | -8.18 | -15.6 | -0.8 | -0.34 | 0.63 | 0.04 | 0.2 |
| TPE | Random | 40 | 9 | -9.08 | -15.41 | -2.15 | -0.42 | 0.73 | 0.01 | 0.06 |
| SMAC | Random | 40 | 9 | -0.9 | -9.11 | 7.46 | -0.03 | 0.5 | 0.83 | 1 |

### Optimizer Convergence Trajectory Summary

**Table 11** A table summarizing the convergence trajectory of the normalized regrets of each optimizer at each family at trial/query 20 and 40. Values are in the form median [Q1, Q3] through the incumbent normalized regret %.

| Surface Family | Query | BOGP | TPE | SMAC | Random Search |
|---|---|---|---|---|---|
| 3optima | 20 | 77.3 [12.8, 99.3] | 69.0 [44.1, 90.0] | 81.1 [53.4, 99.6] | 69.0 [42.9, 93.1] |
| 3optima | 40 | 29.7 [4.8, 72.2] | 41.7 [22.8, 59.6] | 45.0 [18.0, 85.9] | 54.6 [29.2, 71.7] |
| broad_optima | 20 | 38.8 [9.7, 73.2] | 28.7 [9.5, 52.7] | 33.4 [15.4, 63.0] | 48.0 [14.3, 57.1] |
| broad_optima | 40 | 16.0 [5.3, 40.2] | 7.5 [2.8, 11.7] | 10.8 [2.8, 26.4] | 17.4 [6.5, 34.2] |
| heavy_tailed_noise | 20 | 88.8 [28.4, 100.0] | 83.7 [61.4, 100.0] | 93.1 [53.0, 99.9] | 96.0 [61.7, 100.0] |
| heavy_tailed_noise | 40 | 44.6 [7.0, 84.8] | 62.2 [32.0, 80.8] | 55.9 [30.9, 99.8] | 67.2 [43.0, 91.6] |
| heteroscedastic_noise | 20 | 88.8 [27.8, 99.8] | 82.9 [61.4, 99.6] | 91.1 [58.7, 99.9] | 96.6 [61.7, 100.0] |
| heteroscedastic_noise | 40 | 42.5 [8.2, 87.8] | 35.9 [19.6, 86.5] | 51.0 [28.7, 98.7] | 67.2 [44.2, 92.8] |
| quick_rise_slow_fall | 20 | 88.0 [23.6, 100.0] | 79.1 [61.4, 97.2] | 93.1 [59.8, 99.9] | 94.9 [61.7, 100.0] |
| quick_rise_slow_fall | 40 | 37.1 [8.4, 86.6] | 41.8 [21.8, 65.3] | 52.9 [21.8, 98.5] | 67.2 [44.2, 90.5] |
| sharp_optima | 20 | 100.0 [100.0, 100.0] | 100.0 [100.0, 100.0] | 100.0 [100.0, 100.0] | 100.0 [100.0, 100.0] |
| sharp_optima | 40 | 100.0 [88.9, 100.0] | 100.0 [100.0, 100.0] | 100.0 [88.3, 100.0] | 100.0 [96.8, 100.0] |
| slow_rise_quick_fall | 20 | 93.8 [68.6, 98.4] | 92.5 [79.1, 99.9] | 93.4 [79.3, 100.0] | 81.2 [53.3, 99.9] |
| slow_rise_quick_fall | 40 | 71.4 [5.8, 94.6] | 59.6 [39.0, 84.4] | 84.6 [34.1, 99.6] | 53.2 [36.0, 81.1] |
| symmetric | 20 | 100.0 [89.4, 100.0] | 100.0 [96.0, 100.0] | 100.0 [84.9, 100.0] | 100.0 [92.4, 100.0] |
| symmetric | 40 | 99.9 [65.8, 100.0] | 100.0 [80.6, 100.0] | 98.1 [66.1, 100.0] | 90.9 [66.0, 100.0] |
| unique_parameter_interaction | 20 | 83.2 [43.0, 95.5] | 72.5 [46.2, 97.2] | 87.0 [46.4, 99.9] | 77.0 [49.1, 95.8] |
| unique_parameter_interaction | 40 | 61.5 [25.9, 78.5] | 27.9 [12.9, 44.4] | 66.5 [35.7, 95.4] | 60.1 [26.6, 76.1] |

Although BOGP won more families in that table, BOGP's median family profile regret is 60.48, while TPE's median family profile regret is 45.69. Nonetheless, BOGP has a higher probability of reaching the prespecified tolerance, at 0.17, compared to TPE's 0.09.

## Discussion

BOGP performed the strongest among 5 of the optimizer families: namely, the 3 optima family, the heavy-tailed noise family, the sharp optima family, the symmetric family, and the heteroscedastic family. TPE performed strongest in 3 families: the broad optima family, the unique interaction family, and the quick rise slow fall family. Random search performed best in the slow rise quick fall family. However, first, the omnibus evidence was mixed: the Friedman test suggested a difference ($p = .049$), whereas the paired permutation test did not. After the pairwise comparison with Holm's correction, no optimizers were considered statistically different from each other.

The optimizer results across the different response-surface families highlight seemingly poor optimization quality, due to their high normalized regret scores. On top of this, the high standard deviation of the optimizers in certain response surface families, such as the heteroscedastic family, also highlights consistency issues among the Bayesian optimizers. On the other hand, some optimizers performed exceptionally well in certain families, with TPE demonstrating surprising accuracy and small variability in finding the lowest normalized regret scores in the broad optima family.

There were some limitations in this study. For one, the noise added to the noise-free score was not truncated if the resulting value went over 100 or below 0. This thus could've created issues regarding the accuracy and realism of parameter selection in the optimizers.

Also, this purely synthetic simulation at times necessitated more arbitrarily chosen values due to the lack of empirical data in this field, such as the magnitude of the random noise applied to the calculated scores. Basing this function on more empirical data could prevent unhelpful results such as the sharp optima family, where each optimizer was mostly unable to get close to the prespecified normalized regret tolerance of 5 or below. This computational experiment's findings would be made stronger if more data on taVNS effectiveness were available to guide the synthetic simulation. Another strong future idea to pursue is a computational project designed solely on empirical taVNS data. More empirical pupillometry and HRV data would create more accurate surface response families, and more accurate patient parameter variability and measurement noise.

Hyperparameter tuning, which was not implemented in this simulation, could've also improved optimizer effectiveness. For example, under TPE's default configuration, 20 of the trials are simply random samples, while the remaining 20 queries actually apply the TPE optimizer. It would be interesting to see how changing that initial random sample hyperparameter in the optimizers could change their performance

and convergence trajectories.

If this experiment were to be done again, a reexamination of ideal indirect taVNS effectiveness indicators would be done, as pupillometry and heart rate variability may show potential issues in their reliability. A search would be done for similarly designed taVNS clinical studies so their findings could help make the experiment empirically based.

Bayesian optimizer application should be explored in clinical trials of taVNS. Implementing these optimizers in taVNS may in the future exert a positive effect on the many diseases taVNS is under research for use in, such as depression and Alzheimer's disease. With more empirical taVNS data, machine learning optimizers could possibly incorporate patterns of prior taVNS tuning to become even more efficient in optimizing taVNS.

## Acknowledgements

Google Colab and AI were used in the making of the sequential model-based optimization experiment script. I thank Dr Ti'Air Riggins, PhD, for fostering my interest in neuroengineering, her support, and proofreading.

## References

1 Z. Chen and K. Liu, Mechanism and applications of vagus nerve stimulation. Current Issues in Molecular Biology. Vol. 47, pg. 122, 2025 https://doi.org/10.3390/cimb47020122.

2 Y. Bu, A. Liang, B. U. Hoffman, D. M. Schiehser, O. Case, A. Simmons, R. Klaming, A. Gottfried-Blackmore, R. K. Mittal, C. Puleo, H. Lim and I. Lerman, A review of vagus nerve stimulation for disease: comprehensive theory and evidence for mechanisms of action. Comprehensive Physiology. Vol. 16, pg. e70109, 2026 https://doi.org/10.1002/cph4.70109.

3 R. Kamoga, G. Z. Rukundo, S. Kalungi, W. Adriko, G. Nakidde, C. Obua, J. Obongoloch and A. O. Ihunwo, Vagus nerve stimulation in dementia: a scoping review of clinical and pre-clinical studies. AIMS Neuroscience. Vol. 11, pg. 398–420, 2024 https://doi.org/10.3934/Neuroscience.2024024.

4 J. Sun, D. Gao, J. Zhang, H. Liu, Y. Zhou, C. Ma, X. Zhang, S. Liu, J. Zhao, C. Hao and X. Xiao, Transcutaneous auricular vagus nerve stimulation for major depressive disorder: current evidence and future research directions. Frontiers in Psychiatry. Vol. 17, pg. 1788611, 2026 https://doi.org/10.3389/fpsyt.2026.1788611.

5 M. Vargas-Caballero, H. Warming, R. Walker, C. Holmes, G. Cruickshank and B. Patel, Vagus nerve stimulation as a potential therapy in early alzheimer's disease: a review. Frontiers in Human Neuroscience. Vol. 16, pg. 866434, 2022 https://doi.org/10.3389/fnhum.2022.866434.

6 B. W. Badran, A. B. Yu, D. Adair, G. Mappin, W. H. DeVries, D. D. Jenkins, M. S. George and M. Bikson, Laboratory administration of transcutaneous auricular vagus nerve stimulation (tavns): technique, targeting, and considerations. Journal of Visualized Experiments: JoVE. 2019 https://doi.org/10.3791/58984.

7 S. L. Thompson, G. H. O'Leary, C. W. Austelle, E. Gruber, A. T. Kahn, A. J. Manett, B. Short and B. W. Badran, A review of parameter settings for invasive and non-invasive vagus nerve stimulation (vns) applied in neurological and psychiatric disorders. Frontiers in Neuroscience. Vol. 15, pg. 709436, 2021 https://doi.org/10.3389/fnins.2021.709436.

8 P. Atanackov, J. Peterlin, M. Derlink, U. Kovačič, N. Kejžar and F. F. Bajrović, The acute effects of varying frequency and pulse width of transcutaneous auricular vagus nerve stimulation on heart rate variability in healthy adults: a randomized crossover controlled trial. Biomedicines. Vol. 13, pg. 700, 2025 https://doi.org/10.3390/biomedicines13030700.

9 A. D. Farmer, A. Strzelczyk, A. Finisguerra, A. V. Gourine, A. Gharabaghi, A. Hasan, A. M. Burger, A. M. Jaramillo, A. Mertens, A. Majid, B. Verkuil, B. W. Badran, C. Ventura-Bort, C. Gaul, C. Beste, C. M. Warren, D. S. Quintana, D. Hämmerer, E. Freri, E. Frangos, E. Tobaldini, E. Kaniusas, F. Rosenow, F. Capone, F. Panetsos, G. L. Ackland, G. Kaithwas, G. H. O'Leary, H. Genheimer, H. I. L. Jacobs, I. V. Diest, J. Schoenen, J. Redgrave, J. Fang, J. Deuchars, J. C. Széles, J. F. Thayer, K. More, K. Vonck, L. Steenbergen, L. C. Vianna, L. M. McTeague, M. Ludwig, M. G. Veldhuizen, M. D. Couck, M. Casazza, M. Keute, M. Bikson, M. Andreatta, M. D'Agostini, M. Weymar, M. Betts, M. Prigge, M. Kaess, M. Roden, M. Thai, N. M. Schuster, N. Montano, N. Hansen, N. B. Kroemer, P. Rong, R. Fischer, R. H. Howland, R. Sclocco, R. Sellaro, R. G. Garcia, S. Bauer, S. Gancheva, S. Stavrakis, S. Kampusch, S. A. Deuchars, S. Wehner, S. Laborde, T. Usichenko, T. Polak, T. Zaehle, U. Borges, V. Teckentrup, V. K. Jandackova, V. Napadow and J. Koenig, International consensus based review and recommendations for minimum reporting standards in research on transcutaneous vagus nerve stimulation (version 2020). Frontiers in Human Neuroscience. Vol. 14, pg. 568051, 2021 https://doi.org/10.3389/fnhum.2020.568051.

10 A. R. Upadhye, C. Kolluru, L. Druschel, L. A. Lababidi, S. S. Ahmad, D. M. Menendez, O. N. Buyukcelik, M. L. Settell, S. L. Blanz, M. W. Jenkins, D. L. Wilson, J. Zhang, C. Tatsuoka, W. M. Grill, N. A. Pelot, K. A. Ludwig, K. J. Gustafson and A. J. Shoffstall, Fascicles split or merge every ∼560 microns within the human cervical vagus nerve. Journal of Neural Engineering. Vol. 19, pg. 054001, 2022 https://doi.org/10.1088/1741-2552/ac9643.

11 L. Wang, Y. Wang, Y. Wang, F. Wang, J. Zhang, S. Li, M. Wu, L. Li and P. Rong, Transcutaneous auricular vagus nerve stimulators: a review of past, present, and future devices. Expert Review of Medical Devices. Vol. 19, pg. 43–61, 2022 https://doi.org/10.1080/17434440.2022.2020095.

12 L. Yan, H. Li, Y. Qian, J. Zhang, S. Cong, X. Zhang, L. Wu, Y. Wang, M. Wang and T. Yu, Transcutaneous vagus nerve stimulation: a new strategy for alzheimer's disease intervention through the brain-gut-microbiota axis? Frontiers in Aging Neuroscience. Vol. 16, pg. 1334887, 2024 https://doi.org/10.3389/fnagi.2024.1334887.

13 L. Wernisch, T. Edwards, A. Berthon, O. Tessier-Lariviere, E. Sarkans, M. Stoukidi, P. Fortier-Poisson, M. Pinkney, M. Thornton, C. Hanley, S. Lee, J. Jennings, B. Appleton, P. Garsed, B. Patterson, W. Buttinger, S. Gonshaw, M. Jakopec, S. Shunmugam, J. Mamen, A. Tukiainen, G. Lajoie, O. Armitage and E. Hewage, Online bayesian optimization of vagus nerve stimulation. Journal of Neural Engineering. Vol. 21, pg. 026019, 2024 https://doi.org/10.1088/1741-2552/ad33ae.

14 J. Chen, Y. Ke, G. Ni, S. Liu and D. Ming, Tonic and event-related phasic transcutaneous auricular vagus nerve stimulation alters pupil responses in the change-detection task. Neuromodulation: Technology at the Neural Interface. Vol. 28, pg. 652–662, 2025 https://doi.org/10.1016/j.neurom.2025.01.003.

15 J. A. Clancy, D. A. Mary, K. K. Witte, J. P. Greenwood, S. A. Deuchars and J. Deuchars, Non-invasive vagus nerve stimulation in healthy humans reduces sympathetic nerve activity. Brain Stimulation. Vol. 7, pg. 871–877, 2014 https://doi.org/10.1016/j.brs.2014.07.031.

16 V. Wolf, A. Kühnel, V. Teckentrup, J. Koenig and N. B. Kroemer, Does transcutaneous auricular vagus nerve stimulation affect vagally mediated


© The National High School Journal of Science 2026

heart rate variability? a living and interactive bayesian meta-analysis. Psychophysiology. Vol. 58, pg. e13933, 2021 https://doi.org/10.1111/psyp.13933.

17 O. Sharon, F. Fahoum and Y. Nir, Transcutaneous vagus nerve stimulation in humans induces pupil dilation and attenuates alpha oscillations. The Journal of Neuroscience. Vol. 41, pg. 320–330, 2021 https://doi.org/10.1523/JNEUROSCI.1361-20.2020.

18 S. Vespa, L. Stumpp, G. Liberati, J. Delbeke, A. Nonclercq, A. Mouraux and R. E. Tahry, Characterization of vagus nerve stimulation-induced pupillary responses in epileptic patients. Brain Stimulation. Vol. 15, pg. 1498–1507, 2022 https://doi.org/10.1016/j.brs.2022.11.002.

19 M. Keute, M. Demirezen, A. Graf, N. G. Mueller and T. Zaehle, No modulation of pupil size and event-related pupil response by transcutaneous auricular vagus nerve stimulation (tavns). Scientific Reports. Vol. 9, pg. 11452, 2019 https://doi.org/10.1038/s41598-019-47961-4.

20 I. Pervaz, L. Thurn, C. Vezzani, L. Kaluza, A. Kühnel and N. B. Kroemer, Does transcutaneous auricular vagus nerve stimulation alter pupil dilation? a living bayesian meta-analysis. Brain Stimulation. Vol. 18, pg. 148–157, 2025 https://doi.org/10.1016/j.brs.2025.01.022.

21 C. Vezzani, R.-M. Breakspear, L. Thurn, U. Ettinger, A. Kühnel and N. B. Kroemer, Effects of non-invasive vagus nerve stimulation on pupil dilation are dependent on sensory matching. iScience. Vol. 29, pg. 114795, 2026 https://doi.org/10.1016/j.isci.2026.114795.

22 B. Shahriari, K. Swersky, Z. Wang, R. P. Adams and N. D. Freitas, Taking the human out of the loop: a review of bayesian optimization. Proceedings of the IEEE. Vol. 104, pg. 148–175, 2016 https://doi.org/10.1109/JPROC.2015.2494218.

23 J. Snoek, H. Larochelle and R. Adams, Practical bayesian optimization of machine learning algorithms. in Advances in Neural Information Processing Systems (eds F. Pereira, C. J. Burges, L. Bottou & K. Weinberger) Vol. 25 pg. 2951–2959, Curran Associates, Inc., 2012.

24 J. Bergstra, R. Bardenet, Y. Bengio and B. Kégl, Algorithms for hyper-parameter optimization. in Advances in Neural Information Processing Systems Vol. 24 pg. 2546–2554, Curran Associates, Inc., 2011.

25 F. Hutter, H. H. Hoos and K. Leyton-Brown, Sequential model-based optimization for general algorithm configuration. in Learning and Intelligent Optimization (ed. C. A. C. Coello) Vol. 6683 pg. 507–523, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011 https://doi.org/10.1007/978-3-642-25566-3_40.

26 S. Ament, S. Daulton, D. Eriksson, M. Balandat and E. Bakshy, Unexpected improvements to expected improvement for bayesian optimization. in Advances in Neural Information Processing Systems (eds A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt & S. Levine) Vol. 36 pg. 20577–20612, Curran Associates, Inc., 2023 https://doi.org/10.52202/075280-0904.

27 R. R. Souza, N. M. Robertson, C. K. McIntyre, R. L. Rennaker, S. A. Hays and M. P. Kilgard, Vagus nerve stimulation enhances fear extinction as an inverted-u function of stimulation intensity. Experimental Neurology. Vol. 341, 2021 https://doi.org/10.1016/j.expneurol.2021.113718.

28 M. S. Borland, W. A. Vrana, N. A. Moreno, E. A. Fogarty, E. P. Buell, P. Sharma, C. T. Engineer and M. P. Kilgard, Cortical map plasticity as a function of vagus nerve stimulation intensity. Brain Stimulation. Vol. 9, pg. 117–123, 2016 https://doi.org/10.1016/j.brs.2015.08.018.

29 B. W. Badran, O. J. Mithoefer, C. E. Summer, N. T. LaBate, C. E. Glusman, A. W. Badran, W. H. DeVries, P. M. Summers, C. W. Austelle, L. M. McTeague, J. J. Borckardt and M. S. George, Short trains of transcutaneous auricular vagus nerve stimulation (tavns) have parameter-specific effects on heart rate. Brain Stimulation. Vol. 11, pg. 699–708, 2018 https://doi.org/10.1016/j.brs.2018.04.004.

© The National High School Journal of Science 2026