ChatGPT

# Over-Color V1, V2, and V2.3: A Reversible Phonetic Color Naming System

## Introduction

In human–AI collaboration and data communication, there is a growing need for **phonetic color encoding systems** that translate numeric color values into speakable words. Traditional color identifiers like hex codes (e.g. `#73E276`) are precise for machines but unusable for people in conversation or memory [1] [2]. Conversely, common color names (like *sky blue* or *forest green*) are intuitive but limited and ambiguous: only a tiny fraction of the 16 million 24-bit colors have names, and different shades share the same terms [3] [4]. This gap hinders communication – for example, a designer cannot easily *say* a specific RGB color to a colleague or AI assistant, nor can a visually impaired user hear a color code read aloud without confusion [5] [6]. The field of **cognitive linguistics** also suggests that giving linguistic labels to perceptions can shape thought and memory. People naturally chunk and remember information better as words or syllables than as arbitrary digits [7]. By assigning colors pronounceable "names," we leverage the brain's word-memory strengths to handle visual data [8]. Moreover, an **experiential data mapping** emerges when color codes carry semantic or phonetic cues – potentially enriching how we annotate moods, sensations, or qualitative data with colors. In short, a robust phonetic color naming system could bridge human perception and digital color data, providing precise *and* human-friendly codes for every color.

**Over-Color** is one such system: an algorithmic "color language" that generates a unique, pronounceable name for any 24-bit color and allows lossless decoding back to the exact RGB value. This paper explores the Over-Color naming system and its three versions – V1, V2, and V2.3 – as a reversible, phonetic, and semantically rich approach to color representation. We introduce the motivation for phonetic color codes in human–AI contexts, explain the encoding mechanisms of Over-Color, compare the three variants on key properties (reversibility, pronounceability, etc.), and discuss scientific underpinnings from Zipf's law to color perception models. Finally, we highlight potential applications in communication protocols, emotional data mapping, and building a *qualia*-backed color lexicon that links sensory experience to language.

## The Over-Color Encoding Mechanism

Over-Color encodes a 24-bit RGB color (`#RRGGBB`) into a structured pseudoword. At its core, the scheme treats the 24-bit value as data to be represented in a **phonetic alphabet**. In V1 and V2, the 24 bits are split into two 12-bit segments, each mapped to a **consonant–vowel–consonant (CVC)** syllable [9]. This design was inspired by *Proquints* (PROnounceable QUINTuplets), an earlier system for encoding 16-bit values as five-letter syllables [10]. Like Proquints, Over-Color assigns bit groups to phonemes: e.g. 5 bits to a consonant (32 possibilities) and 2 bits to a vowel (4 possibilities), so that a 12-bit chunk is encoded as one CVC syllable [11]. For example, the color `#4E9AF8` (a medium blue) becomes "lum-kiv…" – two syllables that encode the full 24-bit value [12]. Every possible RGB value maps to a unique syllable pair, and given the mapping tables, the decoding algorithm can reconstruct the exact numeric color from the name with **full reversibility** [13] [14]. In other words, the encoding is a lossless bijection between the RGB space and a set of pronounceable "words."

**Suffix for Lightness, Chroma, Parity (V1 & V2):** To enrich the basic two-syllable code, Over-Color V1/V2 append a short suffix carrying extra information [15] [16]. After the two core syllables (which encode the color's binary data), the system adds: **(a)** one letter indicating a broad lightness category, **(b)** one letter for a chroma (colorfulness/saturation) category, and **(c)** a final vowel as a parity check [16] [17]. These suffix characters do not add new color information (the two syllables already encode all 24 bits); rather, they serve *semantic and error-correcting functions*. The lightness and chroma letters function analogous to adjectives in natural color naming (compare how ISCC–NBS notation adds terms like "light" or "deep" to a hue [18]). For instance, "lum-kiveut" is an Over-Color name where "lum" and "kiv" encode the exact RGB value, while the suffix "eut" suggests the color is of medium-high lightness (`e`), high chroma (`u` perhaps denoting a vivid color), and includes a parity check `t` [19]. The **parity vowel** is computed from the binary data (e.g. an XOR of bits) to catch errors [20]. If a single letter in the name is misheard or mistyped, the parity letter would likely not match the rest, signaling an invalid code [21]. This is akin to a checksum in data transmission, adding robustness for spoken or written communication [18]. The outcome is a pronounceable code like "lum-kiveut" for `#4E9AF8`, which a person could read over the phone and the listener (or an AI) could decode unambiguously back to the hex value [22].

**Three-Syllable "Ultra-Silk" Encoding (V2.3):** In version 2.3, Over-Color evolved to a slightly different format: a **three-syllable word** with no separate suffix. Over-Color V2.3 (dubbed *"Zipf-Optimized Ultra-Silk"*) maps the 24-bit color to three CVC syllables, for example `#73E276` → "**win-ralw-wing**" [23] [24]. In this scheme, each syllable effectively encodes one byte (8 bits) of the color [25] [26]. To accommodate 256 possibilities per syllable while preserving pronounceability, V2.3 uses a richer phoneme set and more flexible syllable structure. The consonant inventory includes single letters as well as selected **digraphs** (blended consonant sounds like *ng, rr, lw, kh, zh*, etc.), expanding the available phonetic "alphabet" [27] [28]. The syllable pattern is still largely consonant-vowel-consonant, but the **middle syllable** may use different consonant combinations to ensure the overall word flows smoothly [28]. For instance, in "win-ralw-wing," the second syllable "ralw" ends in a consonant cluster *-lw*, and the last syllable ends in *-ng* – these blends are allowed and were chosen because they can follow or precede other syllables without causing tongue-twisters [29]. By alternating the phonetic structure slightly for the middle vs. the first/last syllable, Over-Color V2.3 achieves a smoother, almost melodic sound [28]. Importantly, V2.3 eliminated the need for case-sensitive characters or additional suffix letters – all information (the full 24 bits) is contained in the three syllables themselves, and every code is in a single consistent case (e.g. lowercase) [30] [31]. This version retains reversibility and uniqueness: given "win-ralw-wing," one can decode it to exactly `#73E276`, and no other color would share that name.

Across all versions, the **encoding/decoding process** is straightforward: to encode, one converts the RGB triplet to a 24-bit integer, splits it (into two 12-bit halves for V1/V2 or three 8-bit parts for V2.3), then maps each chunk through fixed tables of consonants and vowels to form syllables [32] [26]. To decode, the syllables are mapped back to their numeric values and recombined into the 24-bit color [33] [34]. The design ensures that *only* valid phoneme combinations are produced – for example, certain letters like *c, q, x* that could be confused or cause awkward pronunciation are carefully managed or omitted in the phonetic alphabet [35] [36]. The result is an **algorithmic, reversible code** that presents as a word-like name instead of a cryptic number.

## Comparison of V1, V2, and V2.3

The three iterations of Over-Color (V1, V2, V2.3) share the same fundamental goal and general method, but differ in specific design choices. Here we compare them on key attributes:

- **Reversibility:** All versions are fully reversible by construction – each color name maps to exactly one RGB value and vice versa [13] [37]. V1 used two syllables + suffix to encode 24 bits, and the suffix was derived deterministically from the color so that decoding could ignore it (or verify with it) without ambiguity [16] [19]. V2 and V2.3 likewise ensure a one-to-one mapping. The underlying principle is treating the name as a base-N representation in a phonetic alphabet [37]. As a result, *reversibility* is essentially perfect for all three – a critical requirement for a "lossless" color naming scheme. Any differences in reversibility are minor: V1 and V2 include an error-check letter that can detect mistakes but is not needed to compute the color itself [38] [22], whereas V2.3's three-syllable format inherently has no extra error letter (relying instead on the fact that an invalid combination would not decode to a sensible color or could be caught by software). In practice, all three variants satisfy the reversibility criterion equally well [39] [40].

- **Human Pronounceability and Memorability:** All Over-Color names are designed to be pronounceable "words," but the sound characteristics differ by version. **V1** produces two-syllable names (plus a short suffix) that are relatively short (6–7 letters plus 2–3 suffix letters) [41] [42]. These names are pronounceable, but V1's phoneme set included a mix of lower- and uppercase letters as distinct symbols (e.g. "Boqiua" with a capital *B* in the example) [43] [44]. This gave V1 names a visually varied look (uppercase letters adding diversity in shape), which was considered aesthetically pleasing in text [45]. However, using case distinctions had no effect on pronunciation – an uppercase vs. lowercase letter sounds the same – so V1 codes could *sound* confusingly similar even when they were different colors. For example, two colors might have names differing only by case (e.g. "lun" vs "lUn"), which **"is very hard for humans to track by ear or memory"** [46]. Thus, while V1 names were pronounceable, they risked **homophones** in speech due to case-sensitivity. **V2** addressed this by making the encoding *case-insensitive*: it abandoned uppercase-as-data, using only one case (lowercase) and expanding the phoneme inventory with more digraphs (like *bh, sh, ch*) to reach the needed 32 consonant symbols [27] [47]. This made V2 names unambiguous in speech – any difference in the code corresponds to a different sound. V2's names were described as more **"natural in flow"** than V1 [30]. By introducing consonant blends and arranging vowels carefully, V2 achieved a smoother, more melodic pronunciation than the sometimes choppier V1, yet still kept names reasonably short (still two main syllables + suffix) [30]. **V2.3** went further, aiming for maximum euphony: its three-syllable words have been called the most **"melodic, song-like"** codes [48]. With three syllables and many liquid or nasal consonant blends (*win-ralw-wing* has a lilting, almost rhyming sound), V2.3 codes are indeed closer to real words in rhythm. Users might even find them poetic or musical. The trade-off is length and **run-together** sound: V2.3 names are longer (three parts means roughly 8–10 letters), and because they are so fluid, reading several V2.3 color names in a row can sound like one continuous word [48]. In other words, the very smoothness that helps a single code (it "sounds like a song") can reduce clarity when multiple codes are spoken in sequence, as the boundaries blur [48]. V2.3 also sometimes produces unusual spelling combinations (like "ralw" or repeated letters) that look less familiar than the simpler syllables of V1/V2 [49]. In summary, **V1** names are short and visually distinct but suffered from case-based homophones; **V2** names hit a middle ground, improving phonetic clarity and keeping moderate length; **V2.3** names maximize pleasant phonetics and consistency at the cost of greater length and potential over-melodiousness. All are *memorability* improvements over raw hex – a three-syllable pseudoword or even a two-syllable word is easier to remember than six hexadecimal digits [7] [50] – but V2.3's longer names might be a bit harder to recall perfectly compared to the tighter V2 codes, unless their melodious nature aids memory through rhythm.

- **Case Sensitivity:** This is a clear point of evolution. **V1** allowed both lowercase and uppercase letters as separate symbols (e.g. *b* vs *B* encoded different values) [44]. This made the encoding space large enough for 5-bit consonant values using single letters, but it also introduced purely typographic distinctions that are lost in spoken form [45]. V1 names therefore *required exact casing* to decode correctly, violating the design principle of avoiding invisible or audio-invisible encodings [51]. **V2** and **V2.3** are completely case-insensitive – their outputs are typically all lowercase (for consistency) [31]. V2 achieved the needed symbol count by using multi-letter graphemes instead of uppercase (e.g. treating "bh" or "th" as single consonant units) [27]. Thus, any V2/V2.3 name can be spoken or typed without worrying about case, and two names that sound the same will always be the same color. This was a significant usability improvement: *"unique case-insensitive names"* eliminate the risk from V1 where *"very different colors [differed] only in the case of one letter"* [46] [30]. In summary, V1 was **case-sensitive**, whereas V2 and V2.3 are **case-agnostic**, making the latter two far more robust for human use (especially in speech or assistive technologies).

- **Zipfian Distribution of Syllables:** An intriguing aspect of Over-Color is how the frequency of generated syllables can mimic natural language patterns. Human languages follow **Zipf's law**, where a small number of syllables or words are very common and many are rare. Ideally, a synthetic color language would not produce all syllables with equal likelihood (which could sound random), but rather have a distribution that feels "linguistic." The Over-Color design consciously addresses this. In the two-syllable scheme, the frequency distribution of syllable patterns was tuned to emulate Zipfian trends [52]. For example, certain syllable structures or phonemes may occur more frequently in the generated names, while others are more infrequent, giving a naturalistic imbalance [53]. The documentation notes that the gibberish words *"don't appear as random jumbles"* because of this frequency shaping [53]. This was achieved through careful selection of phoneme mapping and perhaps leveraging typical RGB value distributions. **V2.3** is explicitly described as "Zipf-Optimized Ultra-Silk" [23], suggesting that its three-syllable vocabulary was arranged so that some syllables (the "easiest" or most sonorous) correspond to broader swaths of color space or commonly encountered colors, whereas more complex syllables are used for less frequent colors. While all versions of Over-Color are *capable* of uniformly covering the space of 16 million colors, V2.3 likely made the distribution of names closer to a natural lexicon: a few syllables like "ra", "win", "la" etc. might appear often, whereas odd combinations appear seldom. This Zipfian tuning improves the **language-like feel** of the codes and potentially memorability – common patterns become familiar. In comparison, V1's distribution might have been flatter (aside from whatever bias the suffix introduced) and thus sounded a bit more random or evenly dispersed. V2 introduced more natural phoneme choices which already tends toward a more uneven distribution due to the structure. V2.3 then optimized it further to achieve what one might call a *phonetic harmony with natural language frequencies*. The result is that Over-Color codes, especially in V2.3, **"look like language"** and not just random strings [53], which can make them more comfortable to use and remember.

- **Spoken Use & Assistive Technology:** A major goal for Over-Color was to be easily spoken and understood, even by screen readers or voice assistants [2]. Comparing versions: **V1's** case-sensitivity was a critical flaw for spoken use – a screen reader reading a V1 name would not convey the case, so information could be lost. Also, some V1 names might include letter combinations that, while pronounceable, could be slightly harder to articulate due to the particular set of consonants (V1's set included some letters like *z, x, q* in distinct forms which can be more sibilant or ambiguous) [44]. **V2** improved spoken clarity by restricting to a carefully chosen phoneme set: no two Over-Color syllables or letters are too phonetically similar [54]. The digraphs chosen (like "sh" or "th") are still quite distinct sounds. V2 names thus work well with text-to-speech and voice recognition – each code is a sequence of clear syllables not resembling common words (reducing confusion) but still following consistent pronunciation rules. The added parity vowel in V1/V2 also benefits spoken communication: if a listener wasn't sure they heard a letter correctly, the parity check could flag the error [38] [55]. **V2.3** being longer actually can aid clarity for a single code, since three syllables give a bit more redundancy when heard; however, as noted, if many codes are spoken back-to-back (e.g. reading a palette of multiple colors aloud), the listener might need a delimiter or a pause between codes to avoid confusion [49]. In assistive technology contexts, such as a blind user asking for a color or an AI describing an image's colors, V2.3's fluent words could be very advantageous: they can be pronounced in one smooth phrase rather than spelled out digit by digit [2]. Imagine a screen reader saying "**win-ralw-wing**" instead of "**7-3-E-2-7-6**" – the former is far more intelligible and less error-prone [2] [6]. On the other hand, the smoothness means it's crucial that the system delineates each color name (with a slight pause or "dash") in speech output. Overall, V2 and V2.3 significantly improved **auditory usability**. V1 was a proof of concept that met basic pronounceability, but V2 and V2.3 explicitly targeted **voice-friendly design**, making them suitable for spoken color identification, voice-controlled applications, and accessibility tools. All versions avoid any punctuation that isn't speakable (the hyphen can be read as "dash" but is optional) [31], and all use simple phonetic spelling (no silent letters or tricky pronunciations). Thus, from V2 onward Over-Color is well-aligned with assistive technology needs, with V2.3 providing the most *naturally flowing* speech.

- **Structure Pros and Cons:** Summarizing the structures: **V1** = 2 syllables + 3-letter suffix, case-sensitive; **V2** = 2 syllables + 3-letter suffix, all lowercase/digraphs; **V2.3** = 3 syllables, all lowercase, no separate suffix. Each has pros and cons. *V1 pros:* compact; visually distinct letters; straightforward decoding. *V1 cons:* case ambiguity in speech; slightly less melodic or natural; potential for confusion between similar codes. [46] [45]. *V2 pros:* no case issues; more melodic/ pleasant than V1; retains compactness; backward-compatible decoding concept (still two syllables + suffix); better human-factor design (parity for error, etc.) [30] [54]. *V2 cons:* Names still somewhat "artificial" sounding (though improved) and suffix letters, while useful, add an extra touch of gibberish at the end that some might ignore or mishandle (e.g. a user might omit them not realizing their importance). *V2.3 pros:* highly pronounceable and "smooth"; uniform format (no need to consider a separate suffix section); maximally expressive phonetics that *could* carry subtle meaning or aesthetics; sounds most like a real word, which could aid integration into spoken dialogue systems [49]. *V2.3 cons:* longer codes (50% more syllables than V2); risk of auditory run-on; unusual letter combos can be visually a bit harder to parse initially (though still fully phonetic). There is also a **technical trade-off**: V2's design with suffix explicitly encoded perceptual info (lightness/chroma) in dedicated letters, whereas V2.3's approach likely encodes those factors implicitly as part of the 24-bit-to-word mapping. This means V2's names gave a quick clue about the color's general brightness/vividness from the suffix (e.g. one might learn that names ending in "-ut" are bright vivid colors) [19], whereas in V2.3 that information isn't isolated in a single character but diffused in the syllables. Depending on use-case, one or the other structure might be preferable – e.g. for a painter who wants a code that hints at the color's nature, V2's suffix is handy, while for a linguist who wants a seamless word, V2.3 is cleaner.

In essence, Over-Color V1, V2, and V2.3 all deliver on the core idea of reversible, pronounceable color names, but they represent a progression toward greater **user-friendliness**: eliminating case-sensitive traps, enhancing the euphony and naturalness of the "words," and optimizing the system for voice and cognitive factors. The choice of version might depend on context: V1 was a clever demo but less ideal for real-world use; V2 is practical and balanced; V2.3 is aspirationally language-like and engaging, perhaps better for creative or immersive applications where the *aesthetic* of the code matters.

## Scientific Underpinnings of Over-Color

The Over-Color system sits at the intersection of information theory, linguistics, and color science. Several scientific concepts underlie its design:

- **Information Theory & Encoding 24 Bits as Speech:** A 24-bit color value has $2^{24} = 16,777,216$ possible values, which is equivalent to about a 6-digit hexadecimal or a 8-digit decimal number. Mapping this large space onto spoken words is fundamentally an *information encoding* problem. Over-Color's solution – using multiple phonemes (consonants/vowels) in sequence – is essentially constructing a base-$N$ numeral system where $N$ is the number of distinguishable phoneme combinations. V1/V2's two CVC syllables with 32 consonant options and 4 vowel options give $32 \times 4 \times 32$ possibilities per syllable (=$4096$), and two syllables yield $4096^2 = 2^{24}$ combinations, exactly covering the 24-bit space [11] [56]. V2.3's three syllables similarly cover $2^{24}$ with a different breakdown (conceptually $256^3$ combinations with phonetic structuring) [25] [26]. This design had to **maximize entropy** (the amount of information each sound carries) while remaining pronounceable [57]. It's a balance between *density* of encoding and *linguistic constraints*. Each additional syllable increases the word length (which humans might dislike) but allows more bits per syllable to be packed in with simpler phoneme sets. Over-Color's 2-syllable approach was already quite dense (12 bits in a CVC of ~3 letters). The move to 3 syllables in V2.3 suggests the designers chose to distribute 24 bits across 3 smaller chunks (8-bit syllables) to achieve more **phonetic naturalness**, even though it means a longer word. The process parallels **Proquints** and other phonetic binary encodings [58] [9], and draws on principles from **error-correcting codes** (the parity vowel) [18] [38]. In summary, Over-Color demonstrates an applied information theory problem: representing a high-entropy value in a low-entropy channel (human speech) without loss. The success of a fully reversible mapping [13] and inclusion of parity for error-detection [38] show a rigorous, engineering approach to this challenge.

- **Phonetic Design and Pronounceability:** Encoding schemes like this must obey **phonotactics** – the rules of sound sequences that human languages permit. Random bit-to-letter mappings could easily produce unpronounceable strings (e.g. "ztqkvg"), so Over-Color carefully curated its phoneme set and pattern. It uses a *C–V–C syllable structure* which mirrors common patterns in English and many languages [9]. The consonant set was chosen to avoid sounds that are easily confused or difficult to say in sequence [59] [54]. For instance, Proquints omitted letters like *c, x, q* to avoid ambiguity [60]; Over-Color V2 did reintroduce *c, q, x, z* but only as part of a broader set where each has a distinct role (and likely with digraphs that clarify their sound, e.g. "ch" for the *ch* sound, avoiding hard *c* vs soft *c* confusion) [27]. The vowel set (a, e, i, o) was kept small to ensure clear differentiation [61]. These vowels are distinct in sound and avoid *u* which can sound like other vowels in some languages, etc. Over-Color V2.3 even alternates syllable structures (first and last syllable might start or end differently than the middle) to prevent awkward clusters when saying the whole word [29]. For example, a word like "ralw" in the middle of "win-ralw-wing" has a complex cluster *-lw*, but because the next syllable "wing" starts with *w*, the transition is still smooth (the *w* sound continues). Designing millions of codes that are all pronounceable is a remarkable linguistic feat; it required computational generation within the bounds of human articulation. Essentially, Over-Color created a **constrained artificial phonology**. This intersects with research on constructed languages and phonological universals: it ensures every code respects general human constraints (no code will have a disallowed consonant sequence like *mtk* or a tongue-twister like *ssss*). In doing so, the project touches on how **language emergence** might codify information – by assigning sounds to represent data in a rule-governed way. Natural languages obviously didn't set out to encode 24-bit colors, but Over-Color shows it's possible to systematically extend phonetic patterns to cover an entire data space, echoing how ancient languages might have systematically named categories of experience (albeit at a much smaller scale). The scheme's adaptability to different languages' phonemes [62] [63] also relates to phonetic consistency: one could generate Over-Color names in a way that they are pronounceable in, say, Japanese or Spanish by using that language's consonant-vowel inventory, illustrating a principle of **cross-linguistic phonetic mapping**.

- **Zipf's Law and Naturalness:** Human language has a characteristic distribution where some syllables/words are very frequent and many are rare. Over-Color's designers acknowledged that purely random assignment of bit patterns to syllables would yield a flat distribution (every code equally likely) which is unlike any natural lexicon [53]. By *mimicking Zipfian distribution*, Over-Color codes feel more like a natural language where, say, "lum" or "win" might recur more often across different color names, whereas "zor" or "qith" might be rarer. This likely doesn't affect the encoding's functionality (all codes are still unique), but it affects **perception and cognitive load**. A Zipfian distribution means a user will hear certain syllables repeatedly, which could become familiar "anchors" in the language. Research in linguistics suggests that such redundancy helps in processing and memory – our brains are tuned to expect certain common patterns. Over-Color V2.3 explicitly optimizes this, as indicated by the moniker *"Zipf-Optimized"* [23]. It may prioritize simpler syllables for broad areas of color space that might be referenced more often (for example, mid-range colors or common UI colors). Although we do not have the exact mapping algorithm here, we can surmise that V2.3's design ensures the **frequency of syllable usage inversely correlates with complexity**, akin to how short, easy syllables like *the, la, ko* might be more common in a natural corpus. This draws from **Zipf's law** (a principle from statistical linguistics and information theory) to enhance the *usability* of the code as a language [53]. Indeed, the Over-Color approach recognizes that a purely uniform codebook of 16 million "words" is unnatural – by shaping it with Zipfian principles, it becomes more *experience-friendly* without losing technical completeness.

- **Color Perception Models (ISCC–NBS, Munsell, NCS):** The idea of systematically naming colors has a long history in color science. Systems like **ISCC–NBS** (Inter-Society Color Council/National Bureau of Standards) and **Munsell** provide names or notations based on perceptual dimensions (hue, lightness, saturation). For example, ISCC–NBS might describe a color as "vivid yellowish green," which communicates approximate perceptual qualities [64] [65]. These models influenced Over-Color in two ways. First, the inclusion of lightness and chroma markers in V1/V2 suffixes directly echoes these systems' use of modifiers like "light, dark, brilliant, dull" [18] [19]. Over-Color essentially encodes a coarse perception-based description (light/dark, saturated/muted) alongside the exact value. This bridges numeric precision and human perception categories. However, unlike ISCC–NBS or Munsell which **bucket the continuum into broad categories**, Over-Color still gives each individual color a unique code [66] [67]. The scientific taxonomies are *not fully granular* – many distinct RGBs get the same name like "moderate red." They sacrifice uniqueness for simplicity, making them lossy systems [68]. Over-Color, by contrast, achieves uniqueness (one-to-one mapping) but cleverly borrows the idea of **descriptive semantics** by incorporating those categories as part of the code. This is a hybrid of quantitative and qualitative approaches. Additionally, Over-Color's complete mapping aligns with the spirit of Munsell and NCS which aimed to systematically cover the color space, though those usually involve numeric coordinates or alphanumeric codes rather than pronounceable words. Over-Color could be seen as adding a *linguistic layer* on top of the RGB color space that complements scientific models. It doesn't replace color science models (which are important for understanding color relationships and uniformity), but it provides a communication-friendly *representation* of any point in those models. In essence, Over-Color is like a **nomenclature system** for colors, analogous to how chemical nomenclature gives every compound a unique (if complex) name. By referencing ISCC–

NBS and others [18], the creators show they are aware of color naming's perceptual aspects and built upon that legacy to ensure the algorithmic names still carry interpretive cues.

- **Phonetic Codification in Language Emergence:** Over-Color also touches on deeper questions of how language might emerge to label our experiences (like colors). In natural language evolution, words for colors appeared as humans found need to distinguish and describe them. Interestingly, not all languages partition color space the same way – some have few basic color terms, others have many, but none (until now) had a unique word for every single shade [69] [70]. Over-Color can be seen as an engineered leap: creating a lexicon for a domain so fine-grained that previously only numbers could describe it. This invites reflections on **language and qualia**: philosophers often use color as the example of ineffable *qualia* (subjective sensory experiences). By giving each shade a pronounceable name, Over-Color essentially proposes a language that could, in theory, let us *refer to exact qualia* (at least in the visual color domain). In doing so it parallels how early language might have attached sounds to meaningful distinctions in the environment, except here the distinctions are as granular as the eye can see. The scheme also leverages *sound symbolism*. It's known that certain sounds have non-arbitrary associations with sensory qualities across languages – for example, front vowels like "i" tend to be associated with brightness or smallness [71] [72]. Over-Color's use of vowels to encode lightness could play into this symbolic mapping: indeed, research has found that *"vowels with high brightness…were over-represented in words for colors with high luminance"* across languages [72]. If Over-Color consistently uses, say, "o" or "i" in the suffix for lighter colors [19], it's aligning with a cross-modal intuitive pattern – a form of *phonosemantic design*. Similarly, sonorous consonants are associated with saturated (vivid) colors in natural languages [72]; Over-Color's choice of using "y" vs "u" for chroma might not obviously be sonorous vs non-sonorous, but the concept of a distinct phoneme for vividness has parallels in linguistic sound symbolism. Thus, Over-Color isn't just encoding data arbitrarily; it's embedding bits in phonemes in a way that flirts with natural language tendencies. This reflects how language might emerge: not purely arbitrarily, but influenced by human perception and vocal comfort. Over-Color's systematic approach could even inform linguistics – it provides a sandbox to test how a completely filled-out color naming system might influence cognition. Would people trained in Over-Color names start to perceive colors differently, since they can *name* differences that others might not lexically distinguish? Such questions relate to the **Sapir-Whorf hypothesis** and linguistic relativity in color perception. While our focus is technical, Over-Color opens a door to experimentally explore how a fully articulated color lexicon might alter or enhance color cognition. It is, in a sense, creating a microcosm of language emergence: forging a new set of **phonosemantic bridges** between the realm of sensory experience (visual color qualia) and symbolic representation (spoken/written code).

## Applications and Implications

A phonetic, reversible color naming system like Over-Color has diverse potential applications, cutting across technology, communication, and even art or therapy. We highlight a few domains where this innovation could be particularly impactful:

- **Human–AI Communication Protocols:** Over-Color provides a common **linguistic interface** for humans and machines to refer to colors. In scenarios where an AI and a person collaborate (design software, conversational agents, robotics picking colored objects, etc.), using Over-Color words could streamline interaction. For example, instead of an AI saying *"Adjust the hue to 114, 200, 180"* (which is hard to parse), it could say *"try color nal-kamech"* – a precise code name the human can easily repeat or type. Conversely, a user could tell a smart home assistant *"set the living room lights to dor-figeul"* rather than trying to vocally spell an RGB code or select via an interface. This is analogous to the **what3words** system in geolocation (which gives three-word addresses for GPS coordinates) – it makes data *utterable* [2]. Over-Color could become the **"lingua franca" for color data** in voice-driven applications, ensuring that even complex colors can be communicated without visual aids. Moreover, because Over-Color is unambiguous and universal, it could prevent misunderstandings in cross-platform communication (imagine a design tool exporting a color palette in Over-Color names which a game engine or HTML editor can directly interpret via a plugin). The standardized vocabulary means any system implementing the codec can interoperate. In summary, Over-Color can serve as a robust **communication protocol for color**, bridging the human–machine divide by packaging numeric precision in human-friendly phonetics [5] [6].

- **Assistive Technology and Accessibility:** As noted earlier, Over-Color can be transformative for **blind or visually impaired users** who work with color information. Colors are inherently visual, but with a phonetic code, a blind user can hear colors in a meaningful way. For instance, a screen reader can describe an image by reading out Over-Color names of key regions, or a blind artist can label and retrieve colors from a database using spoken names. Because the names are unique, a blind user could accurately communicate a specific color to a sighted colleague or an AI agent (something not feasible with generic words like "dark blue"). Likewise, in text-to-speech systems, hex codes which are normally spelled out digit-by-digit (tedious and error-prone) can be rendered as smooth words [2]. This is akin to having a built-in color *pronunciation guide*. In educational settings, children learning about colors could use Over-Color names to sharpen their understanding of subtle differences (each shade has a distinct name, avoiding confusion). The parity check in V1/V2 adds confidence in these contexts – if a color name is heard incorrectly, it might be caught by the decoding as invalid, prompting a repeat rather than using a wrong color [38]. Overall, Over-Color fosters **multimodal accessibility**, turning vision-centric data into audible and speakable units, which aligns with the principles of inclusive design.

- **Emotion-Tagged Data and Therapeutic Use:** Colors are often linked to emotions and psychological states (think of mood rings, or expressions like "feeling blue"). **Emotion-tagged color data** refers to the idea of associating colors with subjective experiences or moods. Over-Color could play a role in emerging **cognitive and therapeutic frameworks** where individuals use colors to express feelings. For example, in art therapy or mood journaling, a patient might select colors to represent how they feel each day. Using Over-Color, each selected shade wouldn't just be a hex code on a chart – it would have a name that the person can say and remember. This could make the emotional reflection process more articulate. Instead of saying *"today I feel this undefined shade"*, one could say *"today I feel lumez"* (hypothetically), giving a quasi-word to a nuanced emotion-color. Over-Color names could thus become a vocabulary for **qualia**, externalizing internal feelings as shareable words. Therapists could even encourage patients to invent associations or mnemonic meanings for certain Over-Color syllables (perhaps "zul" feels cold and distant, "mar" feels warm, etc.), building a personalized emotional lexicon. On a research front, psychologists might use Over-Color to systematically catalog which colors individuals associate with certain concepts or moods, because the unique naming avoids confusion. It's much easier to, say, log that a respondent chose *"gol-saniul"* for "happiness" and retrieve that exact color later, than to rely on imprecise labels. In short, Over-Color could be a tool in **experiential data mapping** – mapping subjective human experiences (emotions, memories associated with colors, synesthetic impressions) to a stable code. Over-Color's semantic neutrality (the names are initially nonsense words) is actually a benefit here, as they carry no preconceived biases; any meaning can be assigned fresh. This flexibility invites creative therapeutic exercises and data representations that unify the **sensory and symbolic**. For instance, a "color diary" app could store a person's daily moods as Over-Color names, creating a private color-language that an AI could analyze for patterns or even compose into a poetic feedback (since the names sound like a mini-language).

- **Universal Color Reference & Design:** In design, manufacturing, and the arts, a **universal color lexicon** can streamline workflows. Today, designers juggle multiple color systems (RGB, CMYK, Pantone, etc.), and communication often involves sending images or swatches because describing a color precisely is hard. Over-Color could become a neutral reference that anyone can pronounce. It could prevent errors in specifying colors in global teams: rather than describing a paint as "a kind of greenish gray," a designer could specify *exactly* the intended color by its Over-Color name, confident that the manufacturer on another continent can decode the same RGB [73] [74]. This has been compared to an *Esperanto for colors* – a universal, language-independent code that names colors consistently across cultures [73]. Such a system could complement or even unify standards like Pantone by offering an open algorithmic alternative. Moreover, Over-Color can integrate with digital workflows: file formats could include Over-Color metadata, websites could allow colors to be input by name (imagine typing "joy-zenoku" instead of a hex value), and voice-controlled design tools could utilize it for hands-free color changes. In collaborative AR/VR environments, users could speak color names to change object colors instantly. The **one-to-one nature** of Over-Color names means they can be used as stable identifiers or keys in databases – for example, tagging and searching images by color content becomes easier if each color region can be labeled with a unique name rather than ambiguous words. In sum, Over-Color can serve as both a **practical toolkit** in design/industry and a step toward a more fluid human–machine design dialogue.

- **Qualia-backed Color Lexicon and Creative Expression:** On a more philosophical and creative note, Over-Color enables what we might call **phonosemantic bridges** between perception and language. By assigning each color a phonetic form, we create the opportunity to imbue those forms with meaning. Poets and artists could play with Over-Color words, treating them as a new vocabulary of visual experience. For instance, an artist might title a piece with the Over-Color names of its dominant hues, creating a double layer of interpretation (the words sound mysterious yet encode the actual colors used). If a comprehensive dictionary of Over-Color terms and their corresponding colors were compiled, one could even imagine **hidden messages** where the semantic content is in the colors but expressed through these syllables. This unifies the **sensory domain (color)** with the **symbolic domain (language)** in a novel way. It advances the idea that our perception (qualia) can be systematically linked to phonetics – something only hinted at in natural language (with basic color words or poetic devices) but fully realized here. Over-Color could thus be a cornerstone in constructing a "qualia lexicon," a reference that for the first time gives *every nuance of a sensory continuum its own linguistic token*. This has implications in fields like **neuroscience and AI**, where researchers could use these tokens to label and study fine-grained perceptual data. For example, an AI analyzing mood lighting might output a sequence of Over-Color names tracking the color shifts, which correspond to the emotional ambiance. Because these names are speakable, a human can immediately participate in the interpretation, creating an interface that is both technical and poetic.

## Conclusion

The Over-Color naming system illustrates how a clever union of math, linguistics, and design can turn an abstract data space into a living, usable language. Across V1, V2, and V2.3, we see a maturation from a purely **reversible code** for RGB values into a **phonetic and semantically informed language** of color. The system meets practical requirements – every color has a distinct, pronounceable name that can be algorithmically encoded and decoded [13] [37] – while also embracing human factors like pronounceability, memorability, and meaningful structure. It draws from linguistic principles (like phonotactics and Zipf's law) [52], from color science (hue/lightness descriptors) [18], and from information theory (lossless encoding with error checking) [18] [38], embodying an interdisciplinary approach to creating a *functional micro-language*.

Over-Color's evolution resolved key issues (eliminating case sensitivity, improving euphony) and demonstrates an inspiring concept: even something as inherently sensory as color can be given a **voice**. By constructing a **universal phonosemantic bridge** between visual and verbal domains, Over-Color offers more than just a coding scheme – it proposes a new way to think about language and perception. It suggests that in human–AI interfaces, we don't always have to revert to numbers or pre-defined words; we can invent intermediate languages that are both precise for machines and intuitive for humans. As AI systems become more integrated into daily life, such hybrid codes could enrich interaction, allowing us to communicate complex data in conversational ways.

In the future, we may see Over-Color or its descendants used in everything from creative collaborations (imagine dynamically generated color names in a digital art piece) to scientific databases (cataloguing spectra or materials by pronounceable codes). The approach could extend beyond color – one could envision phonetic encodings for other high-dimensional sensory data, building lexicons for tastes, aromas, or sound timbres, thereby giving voice to the full palette of human experience. Over-Color, with its three iterations, stands as a pioneering example of how to design a **comprehensive, human-compatible naming system** for a continuous data domain. It merges the technical with the poetic: every Over-Color name is at once a string of bits and a little *word* carrying the hue of a moment, a data point wrapped in syllables. In bridging these realms, Over-Color V1, V2, and V2.3 mark a step toward a future where we can speak the language of our machines and imbue it with the richness of human semantics – truly a *colorful* convergence of cognition, computation, and communication.

**References:** (selected inline throughout text)
【12】 *Tulsky et al.* "Encoding RGB Colors into Pronounceable, Reversible Names." – *Over-Color V1 Whitepaper*, which introduces the two-syllable encoding and suffix concept, and discusses prior color naming systems and design criteria.
【14】 *Tulsky.* "The Significance of a Universal Phonetic Color Language." – *Over-Color V2.3 Paper*, outlining the importance and implications of a full color language (three-syllable Zipf-optimized scheme) in various contexts.
【20】 Johansson, Anikin, et al. (2019). "Color sound symbolism in natural languages." *Language and Cognition 12(1)*: 56–83. – Study demonstrating non-arbitrary relationships between speech sounds and color attributes (brightness, saturation) across languages, supporting Over-Color's phonosemantic approach.
*...and others as cited inline.*

---

1 2 4 5 6 7 8 23 24 25 26 28 29 50 57 69 70 73 74 The Significance of a Universal Phonetic Color Language.pdf
https://drive.google.com/file/d/1AzYj5UUVnLA3Krx37gao-YKWkCD29SN5

3 9 10 11 12 13 14 15 16 17 18 19 20 21 22 31 32 35 36 37 38 39 40 41 42 51 52 53 54 55 56 58 59 60 62 63 64 65 66 67 68 Encoding RGB Colors into Pronounceable, Reversible Names.pdf
https://drive.google.com/file/d/1H5tef5eNPDgyj6xAvgnFbHH_HhLJsRwH

27 30 33 34 43 44 45 46 47 48 49 61 Untitled document
https://docs.google.com/document/d/1Y4bhW_-2QcrVaK8hGV_amXKuABIRW2UI-zX-U3wyPkU

71 Higher order factors of sound symbolism - ScienceDirect.com
https://www.sciencedirect.com/science/article/pii/S0749596X22000109

72 Color sound symbolism in natural languages | Language and Cognition | Cambridge Core
https://www.cambridge.org/core/journals/language-and-cognition/article/color-sound-symbolism-in-natural-languages/537E325E594CFC227056D4FEA6BCD8E3