

# ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ - ΒΑΘΙΑ ΕΝΙΣΧΥΤΙΚΗ ΜΑΘΗΣΗ

*GenEmoji*

Εργασία 2025

ΠΑΠΑΔΟΠΟΥΛΟΣ ΠΑΝΑΓΙΩΤΗΣ 10697 HMMY

Περιγραφή του Προβλήματος	2
Dataset	2
Εργαλεία	3
Επεξεργασία της εισόδου	3
Επιλογή Emoji - Γενετικός Αλγόριθμος	4
Βελτιστοποίηση Παραμέτρων Γενετικού Αλγορίθμου	5
Δημιουργία Εικόνων Emoji με Ενισχυτική Μάθηση και Νευρωνική Αρχιτεκτονική	10
Αρχιτεκτονική Νευρωνικού Δικτύου και Εκπαίδευση Πράκτορα	11
Πειραματική Αξιολόγηση και Αποτελέσματα	13
Συμπεράσματα και Μελλοντική Εργασία	16
Οδηγίες Εκτέλεσης Πειραμάτων	16

# Περιγραφή του Προβλήματος

Το έργο βασίζεται σε τρεις θεμελιώδεις πυλώνες: την **Ασαφή Λογική**, τους **Γενετικούς Αλγορίθμους** και την **Ενισχυτική Μάθηση** (*Deep Reinforcement Learning*). Η ασαφής λογική χρησιμοποιείται για την ερμηνεία ασαφών ή ημιτελών εκφράσεων στη φυσική γλώσσα του χρήστη, όπως «λίγο λυπημένος» ή «όχι πολύ χαρούμενος», επιτρέποντας στο σύστημα να αποδώσει βαθμούς συμμετοχής (*fuzzy scores*) σε κάθε σχετική λέξη-κλειδί. Οι γενετικοί αλγόριθμοι είναι υπεύθυνοι για την παραγωγή νέων συνδυασμών *emojis*, εξελίσσοντας πληθυσμούς υποψήφιων λύσεων με στόχο τη βέλτιστη προσαρμογή στο επιθυμητό διάνυσμα χαρακτηριστικών. Τέλος, η ενισχυτική μάθηση χρησιμοποιείται στο στάδιο της οπτικής σύνθεσης, όπου ένας πράκτορας εκπαιδεύεται να τοποθετεί τα επιλεγμένα *emojis* πάνω στον καμβά, μαθαίνοντας μέσω επιβράβευσης να δημιουργεί μια οπτικά συνεκτική και σημασιολογικά σωστή απεικόνιση της περιγραφής του χρήστη.

## Dataset

Η βάση δεδομένων του έργου *GenEmoji* έχει κατασκευαστεί ώστε να εξυπηρετεί όλες τις φάσεις της διαδικασίας δημιουργίας νέων *emojis*, από τη γλωσσική κατανόηση μέχρι την οπτική σύνθεση. Κάθε *emoji* συνοδεύεται από τέσσερα βασικά πεδία: το σύμβολο (*emoji character*), τον *Unicode* κωδικό, ένα σύνολο από λέξεις-κλειδιά (*keywords*) που περιγράφουν τα χαρακτηριστικά του, και έναν πίνακα βαθμολογιών ασαφούς λογικής (*fuzzy scores*) που εκφράζει τη σημασιολογική συνάφεια κάθε λέξης-κλειδί με το συγκεκριμένο *emoji*. Τα *fuzzy scores* έχουν εκχωρηθεί είτε χειροκίνητα είτε με τη βοήθεια *LLMs* και αποτελούν τη βάση για την υπολογιστική σύγκριση μεταξύ της επιθυμητής περιγραφής του χρήστη και των υπαρχόντων *emojis*.

Κατά την είσοδο ενός *prompt* από τον χρήστη, το σύστημα εξάγει τις λέξεις-κλειδιά της περιγραφής αναλύοντας την πρόταση. Κάθε λέξη-κλειδί μπορεί να ανήκει σε μία από τις τρεις κατηγορίες:

- Πυρήνας (*core*) και αναφέρεται στο κεντρικό *emoji* με συντελεστή κατηγορίας 1.0
- Τροποποιητής (*modifier*) και εκφράζει κυρίως το συναίσθημα που συνοδεύει τον πυρήνα με συντελεστή 0.9 και
- Εξάρτημα/αξεσουάρ (*accessory*) που συνοδεύει τον πυρήνα με συντελεστή 0.4.

Ο πυρήνας είναι το υποκείμενο της πρότασης, ο τροποποιητής είναι το επίθετο του υποκειμένου και τα εξαρτήματα/αξεσουάρ είναι αντικείμενα της πρότασης. Με αυτόν τον τρόπο δημιουργείται μια ιεραρχία η οποία καθορίζει ποιο είναι το κύριο *emoji*. Για παράδειγμα, η πρόταση “Ένας πικουίνος κρατάει ένα αβοκάντο” δεν θα έπρεπε να έχει το αβοκάντο ως κυρίαρχο στοιχείο στην τελική εικόνα.

Με βάση επιρρημάτων ασαφούς σημασιολογίας (π.χ. «*absolutely*», «*fairly*», «*slightly*») και των παραπάνω συντελεστών κατηγορίας που αναθέτονται με βάση τη γραμματική της πρότασης δημιουργείται ένα διάνυσμα στόχου (*target fuzzy vector*) κάθε λέξη-κλειδί του *prompt* συσχετίζεται με έναν αριθμητικό βαθμό έντασης στο εύρος [0,1]. Το διάνυσμα αυτό εισέρχεται ως στόχος στον γενετικό αλγόριθμο, ο οποίος επιχειρεί να συνθέσει έναν συνδυασμό υπαρχόντων *emojis*, έτσι ώστε το συνολικό τους *fuzzy vector* να προσεγγίζει όσο το δυνατόν περισσότερο τον στόχο.

Τέλος, για την τελική οπτική αναπαράσταση του νέου *emoji*, χρησιμοποιούνται εικόνες σε μορφή PNG, μία για κάθε *emoji* χαρακτήρα. Οι εικόνες αυτές έχουν ληφθεί από την σελίδα *Github* της εταιρίας *X* (πρώην γνωστή ως *Twitter*) και είναι οργανωμένες βάσει του *Unicode* κωδικού κάθε *emoji*, με ειδική επεξεργασία ώστε να αντιστοιχούν σε ονόματα αρχείων. Ο πράκτορας ενισχυτικής μάθησης αναλαμβάνει να τοποθετήσει τις επιλεγμένες εικόνες σε έναν καμβά, αποφασίζοντας τη θέση, το μέγεθος και τη σειρά εμφάνισης κάθε *emoji*, με στόχο τη δημιουργία μιας σύνθετης, οπτικά συνεκτικής εικόνας που να αποδίδει νοηματικά την αρχική περιγραφή του χρήστη.

# Εργαλεία

Βιβλιοθήκη	Ρόλος / Χρήση στο Έργο
gymnasium	Δημιουργία του περιβάλλοντος ενισχυτικής μάθησης ( <i>EmojiCompositionEnv</i> )
stable-baselines3	Εκπαίδευση του πράκτορα PPO για την οπτική σύνθεση emojis
torch	Υλοποίηση του νευρωνικού δικτύου (MLP) στο παρασκήνιο του PPO
numpy	Πίνακες και αριθμητικές πράξεις σε όλο το έργο
pillow	Επεξεργασία εικόνων PNG των emojis και σύνθεση τελικών εικόνων
matplotlib	Δημιουργία διαγραμμάτων ανταμοιβής ανά επεισόδιο για την αξιολόγηση των πρακτόρων
opencv-python	Χρήση για εντοπισμό χαρακτηριστικών (π.χ. <i>HoughCircles</i> ) και περικοπές εικόνας (αν και δεν υλοποιήθηκε πλήρως)
spacy	Ανάλυση φυσικής γλώσσας ( <i>POS tagging, dependency parsing</i> ) για την κατηγοριοποίηση λέξεων σε <i>core, modifier, accessory</i>
scipy	Χρήση σε μετρικές σύγκρισης ή πιθανές ενσωματωμένες συναρτήσεις από την PPO υλοποίηση
tqdm	Εμφάνιση <i>progress bars</i> κατά την εκπαίδευση των πρακτόρων
en_core_web_sm (SpaCy model)	Προεκπαιδευμένο μοντέλο για POS και <i>dependency parsing</i> της αγγλικής γλώσσας

## Επεξεργασία της εισόδου

Η φράση εισόδου που παρέχει ο χρήστης αποτελεί το σημείο εκκίνησης του συστήματος *GenEmoji*. Στόχος του πρώτου σταδίου είναι να μετατραπεί αυτή η φυσική γλώσσα σε μία αριθμητική αναπαράσταση: ένα ασαφές διάνυσμα χαρακτηριστικών, το οποίο θα χρησιμοποιηθεί ως στόχος για τον γενετικό αλγόριθμο.

Αναλυτικά τα βήματα που ακολουθούνται είναι:

### 1. Ορισμός Λεξιλογίου και Ασαφών Γλωσσικών Τροποποιητών

Αρχικά φορτώνεται το λεξιλόγιο λέξεων-κλειδιών από το σύνολο δεδομένων. Το λεξιλόγιο (*vocabulary*) είναι ένα σύνολο λέξεων-κλειδιών που έχει εξαχθεί από το σύνολο δεδομένων των emojis και αντιπροσωπεύει τις οπτικές ή εννοιολογικές έννοιες που μπορεί να περιγράψει ο χρήστης μέσω μιας φράσης εισόδου. Δημιουργείται με τη συλλογή όλων των μοναδικών *tags* και λέξεων-κλειδιών που σχετίζονται με κάθε *emoji* και κάθε λέξη σε αυτό αποτελεί ένα χαρακτηριστικό (*feature*) που μπορεί να περιγράψει κάποιο συναίσθημα (*happy, angry*), αντικείμενο (*sword, hat*), ζώο (*cat, whale*), ή άλλο στοιχείο που εμφανίζεται στις εικόνες των *emojis*. Κατά την επεξεργασία της φράσης του χρήστη, μόνο λέξεις που υπάρχουν στο λεξιλόγιο θεωρούνται έγκυρες για να εισέλθουν στο ασαφές διάνυσμα χαρακτηριστικών. Έτσι διασφαλίζεται ότι το σύστημα λειτουργεί πάνω σε γνωστές και προεπεξεργασμένες έννοιες

Στη συνέχεια, ορίζεται ένας πίνακας που συσχετίζει φραστικούς τροποποιητές (όπως “*very*”, “*slightly*”, “*not really*”) με αριθμητικές τιμές μεταξύ 0 και 1. Αυτές οι τιμές αποτυπώνουν την ένταση ή βεβαιότητα με την οποία ο χρήστης εκφράζει ένα χαρακτηριστικό.

## 2. Συντακτική Ανάλυση με το *SpaCy*

Η φράση περνά από το μοντέλο φυσικής γλώσσας *SpaCy*, το οποίο αναγνωρίζει το συντακτικό ρόλο κάθε λέξης (π.χ. ουσιαστικό, ρήμα, επίθετο, αντικείμενο πρόθεσης κ.λπ.). Αυτό μας επιτρέπει να ομαδοποιήσουμε τις λέξεις σε:

- *core*: βασικά ουσιαστικά (κύριο νόημα),
- *modifier*: επίθετα και επιρρήματα (χαρακτηριστικά ή εντάσεις),
- *accessory*: δευτερεύοντα στοιχεία,
- *background*: το φόντο που μπορεί να υπάρχει στη συνολική εικόνα.

## 3. Εντοπισμός Λέξεων-Κλειδιών

Η φράση διασπάται σε λέξεις και προσδιορίζονται οι λέξεις-κλειδιά που ανήκουν στο λεξιλόγιο. Οι τροποποιητές εντοπίζονται πριν από κάθε λέξη-κλειδί ώστε να προσδιοριστεί η ακριβής βαρύτητά της. Για παράδειγμα, η φράση “*a little angry*” θα αποδώσει στο “*angry*” τιμή 0.3 αντί για 1.0.

## 4. Ανάθεση Τιμών Fuzzy

Για κάθε λέξη-κλειδί, η τελική τιμή προκύπτει από τον πολλαπλασιασμό της τιμής του τροποποιητή με τον συντελεστή βαρύτητας της κατηγορίας της (*core*=1.0, *modifier*=0.9, *accessory*=0.4, *background*=0.5). Αν μία λέξη δεν τροποποιείται, λαμβάνει την πλήρη βαρύτητα της κατηγορίας της. Έτσι δημιουργείται το τελικό *fuzzy vector*, που περιέχει κάθε λέξη-κλειδί και τη σχετική σημασία της βάσει συμφραζομένων.

## 5. Παραγωγή Εξόδου

Η έξοδος της διαδικασίας είναι:

- Ένα *fuzzy vector* σε μορφή λεξικού, όπου κάθε λέξη έχει έναν αριθμό μεταξύ 0 και 1.
- Ένα *role map*, που καταγράφει σε ποια κατηγορία ανήκει κάθε λέξη (*core*, *modifier*, *accessory*).
- Ένα *spatial map*, που περιγράφει τις χωρικές σχέσεις μεταξύ των αντικειμένων του *prompt* με τη μορφή τριάδων (υποκείμενο, σχέση, αντικείμενο), όπως για παράδειγμα (“*penguin*”, “*wearing*”, “*cap*”) ή (“*penguin*”, “*holding*”, “*ice*”).

Αυτό το διάνυσμα λειτουργεί ως στόχος για τον γενετικό αλγόριθμο, ο οποίος θα προσπαθήσει να δημιουργήσει ένα σύνολο *emojis* του οποίου τα χαρακτηριστικά προσεγγίζουν όσο το δυνατόν καλύτερα τις ασαφείς απαιτήσεις της φράσης του χρήστη παρόλο που δεν χρησιμοποιήθηκε στην τελική υλοποίηση.

# Επιλογή Emoji - Γενετικός Αλγόριθμος

Για τη δημιουργία ενός νέου *emoji* που να ανταποκρίνεται στο ασαφές διάνυσμα χαρακτηριστικών της περιγραφής του χρήστη, αξιοποιείται ένας γενετικός αλγόριθμος. Ο γενετικός αλγόριθμος έχει ως στόχο να εντοπίσει τον καλύτερο συνδυασμό *emojis* (χρωμόσωμα), έτσι ώστε να προσεγγίσει όσο το δυνατόν ακριβέστερα το επιθυμητό προφίλ λέξεων-κλειδιών της φράσης του χρήστη.

### Αναπαράσταση λύσης

Κάθε υποψήφια λύση (ή χρωμόσωμα) είναι μια λίστα από  $n$  *emojis*. Κάθε *emoji* θεωρείται ένα γονίδιο, και το σύνολο των *emojis* που επιλέγονται ανά χρωμόσωμα δημιουργούν έναν μοναδικό υποψήφιο συνδυασμό.

### Συνάρτηση Καταλληλότητας (*Fitness Function*)

Ο κάθε υποψήφιος συνδυασμός αξιολογείται μέσω μιας σύνθετης συνάρτησης καταλληλότητας (*fitness*), η οποία λαμβάνει υπόψη:

1. *Fuzzy* ομοιότητα μεταξύ του στόχου (*target vector*) και του διανύσματος που παράγεται από το χρωμόσωμα.
2. Κάλυψη Λέξεων-Κλειδιών: αν περιλαμβάνονται λέξεις του *prompt*.
3. Ποικιλομορφία *emojis* εντός του χρωμοσώματος.
4. Ισχυρή αντιστοίχιση: *emojis* που προσφέρουν τιμές  $\geq 80\%$  του στόχου.
5. Αποδοτικότητα: λόγος μεταξύ κάλυψης και αριθμού *emojis*.
6. Ποινή Πλεονασμού: αν επαναλαμβάνονται οι ίδιες λέξεις-κλειδιά από διαφορετικά *emojis*.
7. Ποινή Θορύβου: αν τα *emojis* περιέχουν άσχετες λέξεις.
8. Μπόνους Σχετικότητας: μικρό *bonus* για υποψήφιες λέξεις που είναι γραμματικά σχετικές.

Ο συνολικός βαθμός κυμαίνεται από 0.0 έως 1.0 και κατευθύνει την εξελικτική διαδικασία.

### Πληθυσμός και Εξέλιξη

Ο αρχικός πληθυσμός αποτελείται από *POP\_SIZE* σε πλήθος τυχαία χρωμοσώματα. Για κάθε γενιά:

- Τα 2 καλύτερα χρωμοσώματα διατηρούνται (ελίτ).
- Γίνεται γονιδιακή αναπαραγωγή μέσω τομής (*crossover*) και επιλέγονται γονείς με βάση τουρνουά.
- Με πιθανότητα *MUTATION\_RATE*, κάποιο γονίδιο του απογόνου μεταλλάσσεται.

Μετά από *GENERATIONS* κύκλους, επιστρέφεται το καλύτερο χρωμόσωμα.

### Συνδυασμός Λέξεων

Το επιλεγμένο χρωμόσωμα αντιστοιχεί σε ένα νέο συνδυασμό *emojis*. Κάθε *emoji* συμβάλλει με το δικό του σύνολο λέξεων-κλειδιών και *fuzzy* βαθμολογιών. Ο συνδυασμός αυτών των πληροφοριών επιτρέπει την προοδευτική προσέγγιση του στόχου, αξιοποιώντας τις δυναμικές της εξελικτικής αναζήτησης.

## Βελτιστοποίηση Παραμέτρων Γενετικού Αλγορίθμου

Παρακάτω παρουσιάζεται η πλήρης ανάλυση όλων των παραμέτρων που επηρεάζουν τη λειτουργία και την αξιολόγηση του γενετικού αλγορίθμου. Περιλαμβάνονται τόσο οι βασικές παραμετρικές ρυθμίσεις της εξελικτικής διαδικασίας, όσο και οι συντελεστές που συμμετέχουν στον υπολογισμό της συνάρτησης καταλληλότητας (*fitness*) κάθε χρωμοσώματος. Οι αρχικές τιμές είναι οι εξής:

Παράμετρος	Τιμή	Αιτιολόγηση
POP SIZE	200	Ο πληθυσμός διατηρείται σχετικά μεγάλος ώστε να υπάρχει επαρκής γενετική ποικιλία και να μειωθεί ο κίνδυνος πρόωρης σύγκλισης.

Παράμετρος	Τιμή	Αιτιολόγηση
CHROMOSOME LENGTH	4	Επιτρέπει την εκπροσώπηση έως και τεσσάρων διαφορετικών <i>emojis</i> , διατηρώντας παράλληλα απλότητα στον έλεγχο.
GENERATIONS	10	Ο αριθμός των γενεών επιλέχθηκε ώστε να επιτραπεί η προοδευτική βελτιστοποίηση των λύσεων, χωρίς υπερβολική επιβάρυνση χρόνου.
MUTATION RATE	0.9	Υψηλό ποσοστό μετάλλαξης που προάγει την εξερεύνηση του χώρου λύσεων και αποτρέπει την παγίδευση σε τοπικά άκρα.

Κριτήριο	Βάρος	Περιγραφή και Αιτιολόγηση
Fuzzy Ομοιότητα	0.4	Μετρά την ομοιότητα μεταξύ του διανύσματος στόχου και του διανύσματος του χρωμοσώματος με χρήση σταθμισμένου <i>dot product</i> .
Κάλυψη Λέξεων-Κλειδιών	0.2	Ποσοστό των λέξεων του <i>prompt</i> που καλύπτονται από τα επιλεγμένα <i>emojis</i> .
Ποικιλομορφία	0.1	Αντιπροσωπεύει το ποσοστό διαφορετικών <i>emojis</i> στο χρωμόσωμα, αποτρέποντας την επανάληψη των ίδιων.
Ισχυρή Αντιστοίχιση	0.2	Ποσοστό λέξεων που καλύπτονται από <i>emojis</i> σε βαθμό τουλάχιστον 80% της επιθυμητής τιμής.
Αποδοτικότητα	0.1	Αναλογία μεταξύ κάλυψης και αριθμού <i>emojis</i> . Επιβραβεύει σύντομα αλλά ακριβή χρωμοσώματα.
Ποινή Πλεονασμού	-0.05	Αφαιρεί πόντους όταν το ίδιο <i>keyword</i> εμφανίζεται πολλές φορές σε διαφορετικά <i>emojis</i> .
Μπόνους Σχετικότητας	+ έως 0.1	Μικρό <i>bonus</i> όταν υπάρχουν γραμματικά ή σημασιολογικά συναφή ζεύγη λέξεων (π.χ. “cat” και “animal”).
Ποινή Θορύβου	-0.1	Τιμωρεί την παρουσία άσχετων <i>emojis</i> (χωρίς καμία λέξη στο <i>prompt</i> ). Βελτιώνει τη συνάφεια.

Η δοκιμή θα πραγματοποιηθεί με το prompt “*an angry penguin holds ice and wears a monocle*”. Οι λέξεις κλειδιά και τα αντίστοιχα βάρη που αναζητεί ο γενετικός αλγόριθμος είναι 'monocle': 0.4, 'penguin': 1.0, 'angry': 0.9 και 'ice': 0.4. Για τις παραπάνω τιμές των παραμέτρων το αποτέλεσμα είναι το εξής: [👊, 🧊, 😡, 🐧]. Το 😡 (*angry face*) και το 🧊 (*ice cube*) είναι σωστές και σχετικές επιλογές — ταιριάζουν άμεσα με τις λέξεις του *prompt*. Τα 👊 και 🧊 είναι πλήρως άσχετα *emojis*, χωρίς καμία αντιστοίχιση στις λέξεις-κλειδιά. Επίσης, δεν περιλαμβάνεται κάποιο *emoji* που να σχετίζεται με *penguin* ή *monocle*, παρόλο που είναι κρίσιμες λέξεις (ειδικά το *penguin* που έχει *score* 1.0).

Δοκιμάζω να τροποποιήσω ορισμένες παραμέτρους.

Παράμετρος	Τιμή	Νέα Τιμή	Αιτιολόγηση
POP SIZE	200	450	Μεγαλύτερος πληθυσμός οδηγεί σε μεγαλύτερη ποικιλία και εξερεύνηση
GENERATIONS	10	15	Η αύξηση του αριθμού των γενεών επιτρέπει στον γενετικό αλγόριθμο να εξερευνήσει πληρέστερα τον χώρο λύσεων και να συγκλίνει σε πιο ακριβείς συνδυασμούς <i>emojis</i>
MUTATION RATE	0.9	0.7	Πολύ υψηλή μετάλλαξη μπορεί να ακυρώσει καλές λύσεις. Ένα πιο συντηρητικό ποσοστό θα βοηθούσε.

Με τις νέες παραμέτρους και παραγόμενο χρωμόσωμα: [👊, 🧊, 🐧, 🧊], φαίνεται ότι η προσαρμογή ενίσχυσε τη σταθερότητα και την ποικιλομορφία του αλγορίθμου. Τα *emojis*:

- 👊 καλύπτει το *angry*,
- 🐧 είναι άμεση αντιστοιχία για *penguin*,
- 🧊 σχετίζεται με *ice*,
- 🧊 παραμένει *noise* (θόρυβος).

Παρατηρήσεις:

- Η μείωση της μετάλλαξης βοήθησε στον περιορισμό της τυχαιότητας, ώστε να μην χαλάει η καλή πληροφορία.
- Η αύξηση του πληθυσμού επέτρεψε ευρύτερη εξερεύνηση.
- Η αύξηση γενεών έδωσε χρόνο στον αλγόριθμο να βελτιώσει τους συνδυασμούς.

Φαίνεται οι δοκιμές στις κύριες παραμέτρους του αλγορίθμου να λειτουργούν. Άρα διαφοροποιούνται οι παράμετροι στην συνάρτηση καταλληλότητας.

Τρέχουσα σύνθεση της *fitness function*:

```
score = (  
    0.4 * fuzzy_similarity +  
    0.2 * coverage_ratio +  
    0.1 * diversity +  
    0.2 * strong_bonus +  
    0.1 * efficiency +  
    relevance_bonus  
)  
score -= 0.05 * redundancy_penalty  
score -= 0.1 * noise_ratio
```

Νέα σύνθεση της *fitness function*:

```
score = (  
    0.5 * fuzzy_similarity +           # +10%  
    0.15 * strong_bonus +             # -5%  
    0.15 * coverage_ratio +          # -5%  
    0.1 * diversity +                 # -  
    0.1 * efficiency +                # -  
    relevance_bonus                   # (μέγιστο +0.1)  
)  
score -= 0.1 * redundancy_penalty     # +0.05  
score -= 0.1 * noise_ratio            # -
```

Κριτήριο	Βάρος	Νέο Βάρος	Αιτιολόγηση
Fuzzy Ομοιότητα	0.4	0.5	Είναι ο βασικός δείκτης ευθυγράμμισης με το <i>prompt</i> . Πρέπει να έχει τη μεγαλύτερη βαρύτητα.
Κάλυψη Λέξεων-Κλειδιών	0.2	0.15	Η κάλυψη λέξεων μπορεί να προέλθει και από γενικά σχετικά <i>emojis</i> . Δεν πρέπει να υπερεκτιμάται.
Ποικιλομορφία	0.1	0.1	Προσφέρει σταθερότητα και ισορροπία στην επιλογή χρωμοσωμάτων.
Ισχυρή Αντιστοίχιση	0.2	0.15	Ενισχύει ελαφρώς <i>emojis</i> που περιέχουν παραλλαγές ή συνώνυμα λέξεων του <i>prompt</i> .
Αποδοτικότητα	0.1	0.1	Προσφέρει σταθερότητα και ισορροπία στην επιλογή χρωμοσωμάτων.



Κριτήριο	Βάρος	Νέο Βάρος	Αιτιολόγηση
Ποινή Πλεονασμού	-0.05	-0.1	Ο πλεονασμός λέξεων-κλειδίων (από διαφορετικά <i>emojis</i> ) μειώνει την ποικιλομορφία και την ακρίβεια.
Μπόνους Σχετικότητας	+ έως 0.1	+ έως 0.1	Η μείωση αποτρέπει την υπερεκτίμηση μεμονωμένων ισχυρών αντιστοιχιών.
Ποινή Θορύβου	-0.1	-0.1	Τιμωρεί την παρουσία άσχετων <i>emojis</i> (χωρίς καμία λέξη στο <i>prompt</i> ). Βελτιώνει τη συνάφεια.

Με αυτές τις αλλαγές, ο γενετικός αλγόριθμος καταλήγει σε χρωμοσώματα όπως:

- ['🦵', '🐼', '🤪', '🍷']
- ['🖨️', '🖨️', '🤪', '🐼']
- ['📺', '🏠', '🐼', '😈']
- ['🏠', '🍷', '🐼', '😬']

Είναι εμφανές ότι οι παραπάνω συνδυασμοί καλύπτουν τα κύρια χαρακτηριστικά του *feature vector*, χωρίς όμως να το καλύπτουν εξ' ολοκλήρου. Παρατηρείται ότι κανένας από τους συνδυασμούς δεν εμπεριέχει το *emoji* 🤪. Αυτό μπορεί να οφείλεται είτε στο γεγονός ότι ο αλγόριθμος συναντά το *emoji* αυτό αλλά το απορρίπτει είτε ότι δεν το συναντά ποτέ. Εξαιτίας του γεγονότος ότι το μεγαλύτερο πλήθος των *emoji* βρέθηκαν σωστά (ή αρκετά κοντά), θεωρείται ότι ο αλγόριθμος δεν εξερευνεί αρκετά για να συναντήσει τον κατάλληλο συνδυασμό. Επομένως γίνεται δοκιμή με μεταβλητό ρυθμό μετάλλαξης, ο οποίος δίνεται από τον τύπο:

**MUTATION\_RATE = max(0.1, 0.9 - generation \* 0.01)**

Πράγματι, η τελευταία αλλαγή συνέβαλε στην καλύτερη επιλογή *emoji*. Ωστόσο, η ορθή επιλογή δεν γίνεται με συνέπεια. Λόγω του μεγάλου πλήθους δειγμάτων, δεν είναι πάντα εφικτό ο αλγόριθμος να βρίσκει την βέλτιστη λύση, ειδικά εφόσον τα αρχικά γονίδια επιλέγονται τυχαία. Φυσικά, με κάθε νέα γενιά αυξάνονται οι πιθανότητες να ανακαλυφθεί ο βέλτιστος συνδυασμός.

Αξίζει να σημειωθεί ότι συγκεκριμένα εσφαλμένα *emoji* επαναλαμβάνονται συστηματικά σε διαφορετικές εκτελέσεις του γενετικού αλγορίθμου. Αυτό οφείλεται κυρίως στον τρόπο με τον οποίο λειτουργεί η φυσική επιλογή και η πίεση επιλογής στη διαδικασία της εξέλιξης. Τα συγκεκριμένα αυτά *emoji*, παρότι δεν είναι τα βέλτιστα για το ζητούμενο *prompt*, παρέχουν αρκετά υψηλές τιμές σκορ (*fitness*), κυρίως επειδή ευθυγραμμίζονται εν μέρει με κάποιες από τις λέξεις-κλειδιά του διανύσματος στόχου ή παρουσιάζουν υψηλή *fuzzy* ομοιότητα σε άλλες λέξεις του *prompt*. Ως αποτέλεσμα, “επιβιώνουν” από γενιά σε γενιά, κληρονομούνται σε απογόνους και κυριαρχούν στον πληθυσμό, οδηγώντας τον αλγόριθμο σε τοπικό μέγιστο αντί για το ολικό. Το γεγονός αυτό αναδεικνύει τη σημασία της ποικιλομορφίας του πληθυσμού και της εξερεύνησης μέσω μετάλλαξης, ώστε ο αλγόριθμος να μπορεί να ξεφύγει από τέτοιες παγίδες και να εξετάσει εναλλακτικές, πιο ακριβείς λύσεις.

# Δημιουργία Εικόνων Emoji με Ενισχυτική Μάθηση και Νευρωνική Αρχιτεκτονική

Το τρίτο και πιο απαιτητικό στάδιο του συστήματος *GenEmoji* αφορά τη σύνθεση της τελικής εικόνας, η οποία συνδυάζει τα επιλεγμένα *emojis* με στόχο τη δημιουργία μιας νοηματικά και οπτικά συνεκτικής σκηνής. Η διαδικασία αυτή υλοποιείται μέσω ενός πράκτορα ενισχυτικής μάθησης (*RL agent*), ο οποίος εκπαιδεύεται ώστε να αποφασίζει ποιο *emoji* να τοποθετήσει, πού, με ποιο μέγεθος και σε ποιο βάθος (*layer*) της εικόνας.

## Απόφαση του Πράκτορα – Ενέργειες Ανά Βήμα

Ο πράκτορας λειτουργεί βηματικά, και σε κάθε βήμα παράγει μια ενέργεια υπό μορφή διανύσματος 8 παραμέτρων, που περιγράφει ακριβώς ποια ενέργεια σύνθεσης θα εκτελεστεί:

- *Emoji index*: Επιλογή του *emoji* από τη διαθέσιμη λίστα.
- Θέση ( $x, y$ ): Καθορισμός των κανονικοποιημένων συντεταγμένων τοποθέτησης στον καμβά.
- Κλίμακα (*scale*): Επιλογή μεγέθους (*core* → μεγάλο, *modifier* → μεσαίο, *accessory* → μικρό).
- Επίπεδο (*layer*): Το βάθος τοποθέτησης (*φόντο*, μέσο επίπεδο, προσκήνιο).
- *Color reference*: Επιλογή *emoji* πηγής χρώματος και εφαρμογή του μέσω *tint*.
- *Crop type*: Επιλογή είδους περικοπής χαρακτηριστικών.
- *Stop flag*: Αν είναι ενεργό ( $> 0.5$ ), ο πράκτορας δηλώνει ότι ολοκλήρωσε τη σύνθεση.

## Σύνθεση στον Καμβά – Πρακτική Εφαρμογή

Μόλις ο πράκτορας παράγει μια ενέργεια, το περιβάλλον την εκτελεί ως εξής:

- Ανάκτηση εικόνας: Φόρτωση του αντίστοιχου αρχείου PNG από το *emoji\_to\_path*, βάσει του *Unicode* του *emoji*.
- Μετασχηματισμός μεγέθους: Η εικόνα κλιμακώνεται κατάλληλα ανάλογα με τον ρόλο της.
- Χρωματική επεξεργασία: Αν έχει οριστεί χρωματική αναφορά, εφαρμόζεται *tint* από το χρώμα του *emoji* πηγής.
- Περικοπή χαρακτηριστικών: Αν επιλεγεί *crop*, εφαρμόζεται εντοπισμός στοιχείων (μάτια, πρόσωπο) με χρήση *HoughCircles* και *OpenCV*, και πραγματοποιείται αντικατάσταση.
- Τοποθέτηση στον καμβά: Η επεξεργασμένη εικόνα επικολλάται στις ζητούμενες συντεταγμένες και στο σωστό *layer*. Το σύστημα διατηρεί λίστα *placed\_emojis* με *metadata* για κάθε σύμβολο (θέση, μέγεθος, *layer*), ώστε να αναδομηθεί ο καμβάς με ακρίβεια.

## Ανταμοιβή και Ανατροφοδότηση – Πώς μαθαίνει ο πράκτορας

Η συνάρτηση ανταμοιβής καθοδηγεί τη μάθηση και επιστρέφει έναν αριθμό που εκφράζει την ποιότητα της σύνθεσης. Η ανταμοιβή απονέμεται σε κάθε βήμα και τελικά στο τέλος του επεισοδίου.

Θετικές ενισχύσεις δίνονται όταν:

- Υπάρχει *core emoji* στην εικόνα.
- Το *background* είναι ορθό και στο *layer 0*.
- Τα *accessories* βρίσκονται κοντά στο *core*.
- Ο πράκτορας σταματά εγκαίρως με χρήση της ενέργειας “*stop*”.

Ποινές αποδίδονται όταν:

- Τοποθετούνται ίδια σύμβολα επανειλημμένα.
- Τα μεγέθη ή τα *layers* είναι ακατάλληλα.
- Εμφανίζεται *emoji* που προοριζόταν μόνο για *color reference*.
- Απουσιάζουν απαραίτητοι ρόλοι (*core*, *background*).
- Ο αριθμός ενεργειών είναι υπερβολικά μικρός ή μεγάλος.

## Στόχος της Μάθησης

Ο πράκτορας καλείται να μάθει όχι μόνο ποια *emojis* να επιλέγει, αλλά κυρίως πώς και πού να τα τοποθετεί, ώστε η παραγόμενη εικόνα να αποδίδει οπτικά και νοηματικά τη σκηνή του *prompt*. Η βελτίωση της πολιτικής γίνεται μέσω PPO (*Proximal Policy Optimization*), που επιτρέπει σταθερή και σταδιακή προσαρμογή της συμπεριφοράς, ενισχύοντας τη δημιουργικότητα και την ακρίβεια της σύνθεσης.

# Αρχιτεκτονική Νευρωνικού Δικτύου και Εκπαίδευση Πράκτορα

Το στάδιο αυτό επικεντρώνεται στην εκπαίδευση του πράκτορα ενισχυτικής μάθησης (RL agent), ώστε να τοποθετεί τα επιλεγμένα *emojis* πάνω στον καμβά με τρόπο που να αποδίδει σωστά τη σημασιολογική δομή της φυσικής γλώσσας. Η μάθηση βασίζεται σε ενισχυτική εκπαίδευση τύπου PPO, με πολιτικές που μοντελοποιούνται μέσω νευρωνικών δικτύων τύπου MLP.

## Περιβάλλον Εκπαίδευσης

Ο πράκτορας εκπαιδεύεται στο περιβάλλον *EmojiCompositionEnv*, βασισμένο στο *Gymnasium*, με παρατήρηση τύπου RGB εικόνας (256×256×3) και συνεχές διάνυσμα δράσης 8 διαστάσεων. Η ανταμοιβή καθορίζεται με βάση αισθητικά, χωρικά και σημασιολογικά κριτήρια και επιδιώκει την παραγωγή ποιοτικά συνεκτικών συνθέσεων.

### Επιλογή Αλγορίθμου και Πολιτικής

Για την εκπαίδευση χρησιμοποιήθηκε ο αλγόριθμος *Proximal Policy Optimization* (PPO) από τη βιβλιοθήκη *stable-baselines3*. Ο PPO είναι ένας *state-of-the-art* αλγόριθμος *on-policy*, κατάλληλος για συνεχή χώρο ενεργειών, καθώς επιτρέπει σταθερή και ασφαλή ενημέρωση της πολιτικής. Η πολιτική μοντελοποιείται μέσω ενός πολυεπίπεδου *perceptron* (MLP), το οποίο εκπαιδεύεται να προβλέπει την επόμενη ενέργεια βάσει της παρατήρησης του καμβά.

Η εκπαίδευση έγινε εξολοκλήρου σε CPU, με χρήση 10 επεισοδίων ανά *iteration* και ενσωματωμένη παράμετρο *ent\_coef=0.1* για έλεγχο της εντροπίας της πολιτικής.

### Διαδικασία Εκπαίδευσης

Για κάθε διαφορετική αρχιτεκτονική, ο πράκτορας εκπαιδεύεται στο ίδιο περιβάλλον με τα ίδια *emojis*, *fuzzy vector* και *spatial map*. Η παρατήρηση μετασχηματίζεται με *VecTransposeImage* ώστε να είναι συμβατή με τον αλγόριθμο PPO. Το εκπαιδευμένο μοντέλο αποθηκεύεται ως .zip στον φάκελο *models/{αρχιτεκτονική}* για μεταγενέστερη αξιολόγηση.

### Αρχιτεκτονικές Νευρωνικού Δικτύου

Για την αξιολόγηση της σχέσης μεταξύ της πολυπλοκότητας του νευρωνικού δικτύου και της απόδοσης του πράκτορα, υλοποιήθηκαν και εκπαιδεύτηκαν διαφορετικές αρχιτεκτονικές MLP. Το ζητούμενο είναι να εξεταστεί πώς μεταβάλλεται η επίδοση της πολιτικής καθώς αυξάνεται το βάθος και το εύρος του νευρωνικού δικτύου.

Οι πολιτικές που χρησιμοποιούνται στο πλαίσιο του PPO ορίζονται ως δύο ανεξάρτητα MLPs, ένα για την πρόβλεψη της ενέργειας (*actor*) και ένα για την εκτίμηση της αξίας της κατάστασης (*critic*). Η υλοποίηση έγινε μέσω της *MlpPolicy* της βιβλιοθήκης *stable-baselines3*, χωρίς να χρειαστεί ρητός διαχωρισμός *actor/critic* από την πλευρά του χρήστη. Οι αρχιτεκτονικές που εξετάζονται στον παρόν πειραματισμό περιλαμβάνουν τις εξής δομές:

Όνομα Αρχιτεκτονικής	Δομή Κρυφών Επιπέδων	Χρονικά Βήματα Εκπαίδευσης ( <i>timesteps</i> )	Χρόνος Εκπαίδευσης ( <i>min</i> )	Παρατηρήσεις
mlp_32x2	[32, 32]	8.000	2,5 λεπτά	Ελαφρύ, χρήσιμο για <i>baseline</i>
mlp_64x2	[64, 64]	10.000	4,5 λεπτά	Ελαφρύ, <i>baseline</i>
mlp_128x1	[128]	25.000	19 λεπτά	Χαμηλό βάθος, ίδιο πλάτος με 128x2
mlp_128x2	[128, 128]	30.000	21,5 λεπτά	Σταθερότητα και απόδοση

mlp_256x2	[256, 256]	50.000	1,5 ώρες	Υψηλή χωρητικότητα
mlp_128x3	[128, 128, 128]	60.000	43 λεπτά	Αυξημένο βάθος

Σε κάθε περίπτωση, το ίδιο δίκτυο χρησιμοποιείται τόσο για την πολιτική όσο και για τη συνάρτηση αξίας. Η ενεργοποίηση που εφαρμόζεται σε κάθε επίπεδο είναι η *ReLU*, ενώ δεν χρησιμοποιούνται *convolutional* ή *recurrent* μονάδες, καθώς η φύση του προβλήματος δεν απαιτεί χωρική ή χρονική αλληλεξάρτηση.

Η επιλογή των συγκεκριμένων αρχιτεκτονικών έγινε με στόχο να παρατηρηθεί:

- η ταχύτητα σύγκλισης κάθε μοντέλου,
- η ποιότητα των παραγόμενων εικόνων,
- και η σταθερότητα των ανταμοιβών κατά τη διάρκεια των επεισοδίων δοκιμής.

Επιπλέον, η χρήση διαφορετικού πλήθους χρονικών βημάτων εκπαίδευσης (*timesteps*) ανά αρχιτεκτονική στοχεύει στην αποδοτικότερη αξιοποίηση της χωρητικότητας κάθε δικτύου, με σκοπό να επιτευχθεί επαρκής μάθηση χωρίς *overfitting*.

## Πειραματική Αξιολόγηση και Αποτελέσματα

Η ενότητα αυτή αναλύει την επίδοση των πρακτόρων που εκπαιδεύτηκαν με διαφορετικές αρχιτεκτονικές MLP, με βάση:

- Την ποσοτική απόδοση μέσω *reward*,
- Την σταθερότητα και συμπεριφορά του πράκτορα,
- Και τις παρατηρήσεις που προκύπτουν από τη σύγκριση των αποτελεσμάτων.

### Πειραματική Μεθοδολογία

Κάθε πράκτορας αξιολογήθηκε ανεξάρτητα σε 10 δοκιμαστικά επεισόδια, στα οποία του ζητήθηκε να συνθέσει μια εικόνα βάσει του ίδιου γονιδίου του γεννητικού αλγορίθμου και των ίδιων *fuzzy* χαρακτηριστικών. Στόχος της διαδικασίας ήταν να διαπιστωθεί αν ο πράκτορας έχει όντως μάθει να συνθέτει σκηνές που αποτυπώνουν τις σημασιολογικές και χωρικές απαιτήσεις της φυσικής γλώσσας.

Τα μοντέλα είχαν εκπαιδευτεί εκ των προτέρων και φορτώνονταν από αποθηκευμένα αρχεία .zip. Το περιβάλλον παρέμεινε αμετάβλητο για όλους τους πράκτορες, ώστε η σύγκριση να είναι αντικειμενική. Για κάθε επεισόδιο αποθηκευόταν η τελική εικόνα σε μορφή .png, παρέχοντας έτσι και δυνατότητα ποιοτικής οπτικής αξιολόγησης.

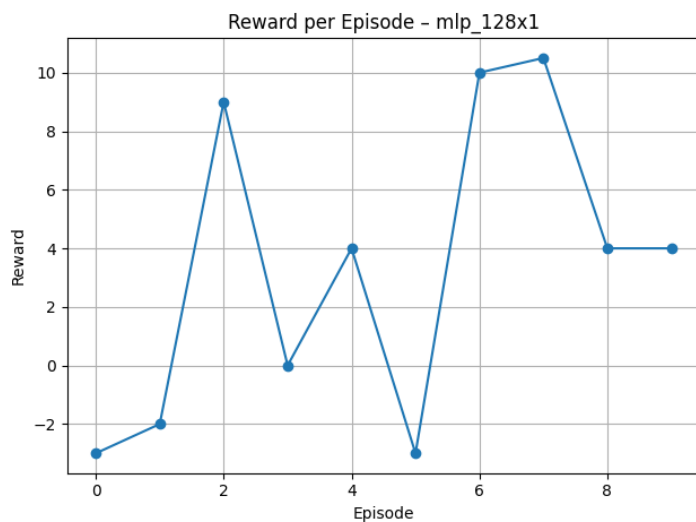
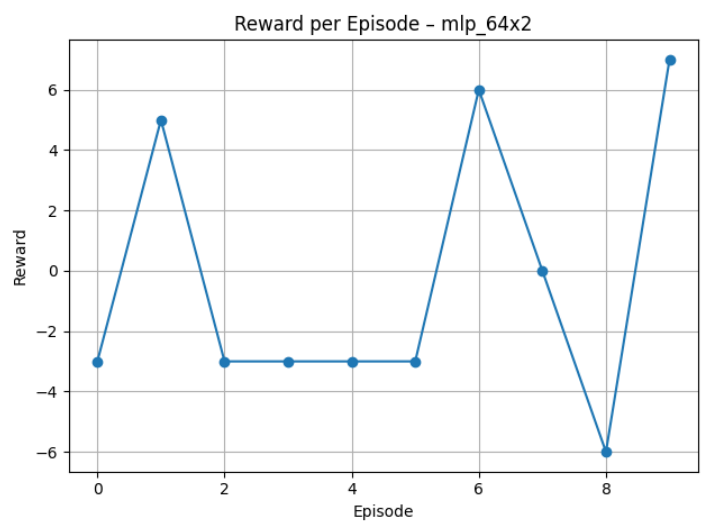
Κατά την αξιολόγηση, χρησιμοποιήθηκε *deterministic=False* στη μέθοδο *model.predict*, ώστε ο πράκτορας να δειγματοληπτεί ενέργειες από την κατανομή που έχει μάθει αντί να επιλέγει πάντα τη μέση ή πιο πιθανή τιμή. Αυτό κρίθηκε απαραίτητο, διότι με ενεργοποιημένο το *deterministic=True*, παρατηρήθηκε εκφυλισμένη συμπεριφορά: ο πράκτορας τοποθετούσε το ίδιο *emoji* επανειλημμένα στην ίδια θέση, χωρίς καμία ποικιλία ή σημασιολογική συνάφεια. Αντίθετα, με *deterministic=False*, η πολιτική του πράκτορα εκφράζει ποικιλία και καταφέρει να αποδώσει πιο φυσικές, συνεκτικές και νοηματικές σκηνές. Αυτό

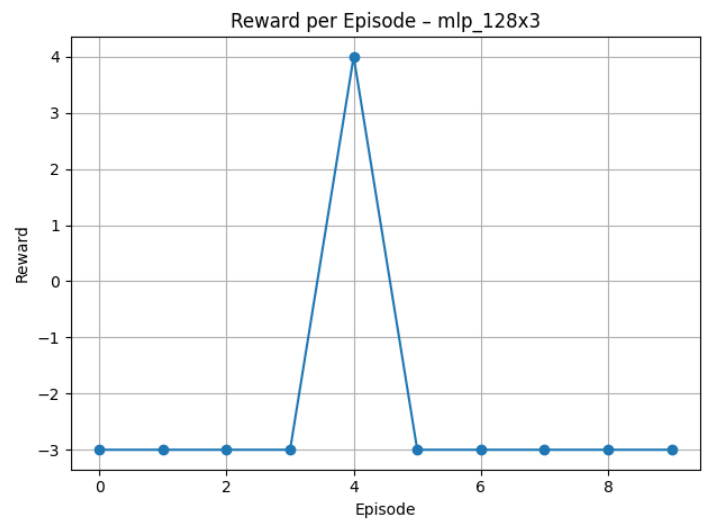
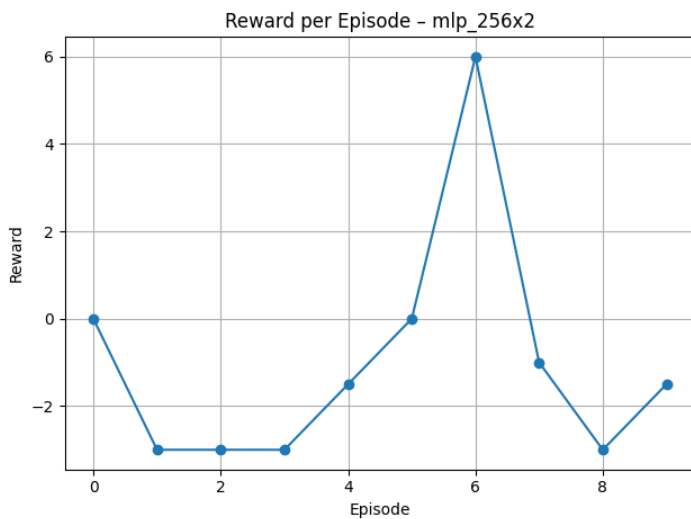
καταδεικνύει ότι, παρόλο που το δίκτυο δεν έχει εκπαιδευτεί επαρκώς ώστε να συγκλίνει σε "αιχμηρή" ντετερμινιστική πολιτική, έχει μάθει μια κατανομή ενεργειών με χρήσιμο και εκφραστικό περιεχόμενο.

### Ανάλυση Ανταμοιβών

Τα αποτελέσματα των ανταμοιβών για κάθε αρχιτεκτονική παρουσιάζονται στον ακόλουθο πίνακα και στο αντίστοιχο διάγραμμα *reward\_plot.png*. Για κάθε μοντέλο, υπολογίστηκε ο μέσος όρος ανταμοιβής στα 10 επεισόδια καθώς και η διακύμανση των επιδόσεων μεταξύ των δοκιμών.

Αρχιτεκτονική	Μέσο Reward	Τυπική Απόκλιση	Ελάχιστο	Μέγιστο
mlp_32x2	-1.25	2.78	-4.0	4.5
mlp_64x2	-0.3	4.6	-6.0	7.0
mlp_128x1	3.35	5.26	-3.0	10.5
mlp_128x2	1.9	6.55	-9.5	12.0
mlp_256x2	-1.0	2.74	-3.0	6.0
mlp_128x3	-2.3	2.21	-3.0	4.0





### Παρατηρήσεις και Συμπεράσματα

Η ανάλυση των επιδόσεων των έξι διαφορετικών αρχιτεκτονικών MLP αναδεικνύει σαφείς διαφοροποιήσεις ως προς τη σταθερότητα και την ποιότητα των ενεργειών του πράκτορα:

- Οι αρχιτεκτονικές `mlp_128x1` και `mlp_128x2` εμφάνισαν τη βέλτιστη συνολική απόδοση, με μέσο *reward* 3.35 και 1.90 αντίστοιχα, καθώς και μέγιστες τιμές ανταμοιβής 10.5 και 12.0. Η συμπεριφορά τους στα αντίστοιχα διαγράμματα είναι έντονα ανοδική σε πολλά επεισόδια, κάτι που υποδηλώνει ικανότητα σύνθεσης σωστής εικόνας με επαναληψιμότητα.
- Η αρχιτεκτονική `mlp_128x2`, αν και παρουσιάζει υψηλό μέγιστο, συνοδεύεται από την υψηλότερη τυπική απόκλιση (6.55), κάτι που αποκαλύπτει αστάθεια στη συμπεριφορά του πράκτορα: σε κάποιες περιπτώσεις αποδίδει εξαιρετικά, ενώ σε άλλες αποτυγχάνει.
- Η αρχιτεκτονική `mlp_128x3`, παρόλο που είναι η πιο βαθιά από τις 128-units, εμφάνισε τη χειρότερη συνολική συμπεριφορά, με μέσο *reward* -2.3 και ελάχιστη μεταβλητότητα. Αυτό μαρτυρά ότι ο πράκτορας δεν κατάφερε να αξιοποιήσει την επιπλέον πολυπλοκότητα και συγκλίνει σε φτωχή πολιτική.
- Οι μικρότερες αρχιτεκτονικές, `mlp_32x2` και `mlp_64x2`, εμφάνισαν μέσες τιμές ανταμοιβής κάτω από το μηδέν, καθώς και μεγάλες αποκλίσεις μεταξύ επεισοδίων. Η `mlp_64x2` παρουσίασε μεγάλη αστάθεια, με *reward* έως και 7.0, αλλά και χαμηλές τιμές κάτω του -6. Αυτό δείχνει ότι οι μικρότερες δομές αποτυγχάνουν να γενικεύσουν.
- Η αρχιτεκτονική `mlp_256x2` είχε παρόμοια απόδοση με τις μικρές, παρουσιάζοντας μέσο *reward* -1.0, παρά την αυξημένη χωρητικότητα. Αυτό ενδέχεται να οφείλεται είτε σε *overfitting* είτε σε αδυναμία αξιοποίησης των επιπλέον παραμέτρων λόγω περιορισμένης εκπαίδευσης.

Συμπερασματικά, προκύπτει ότι η `mlp_128x1` είναι η πλέον ισορροπημένη αρχιτεκτονική όσον αφορά την ποιότητα, τη σταθερότητα και την ικανότητα να αποδώσει συνεκτικές εικόνες. Η `mlp_128x2` εμφανίζει το υψηλότερο δυναμικό, αλλά απαιτεί βελτιώσεις στη σταθεροποίηση της πολιτικής. Οι υπόλοιπες αρχιτεκτονικές είτε δεν κατάφεραν να μάθουν αποδοτική πολιτική, είτε παρουσίασαν υψηλή μεταβλητότητα χωρίς συνέπεια.

Η χρήση του `deterministic=False` ήταν καθοριστική για την αποκάλυψη των πραγματικών ικανοτήτων των πολιτικών, ενώ η απόδοση σε ντετερμινιστική λειτουργία παραμένει φτωχή. Το γεγονός αυτό υποδηλώνει ότι οι πράκτορες έχουν μεν μάθει μία κατανομή ενεργειών που οδηγεί σε σωστές τοποθετήσεις, αλλά δεν έχουν εκπαιδευτεί επαρκώς ώστε να συγκλίνουν σε σαφή αποφασιστικές πολιτικές.

Συνολικά, η αξιολόγηση επιβεβαιώνει την επάρκεια της μεθοδολογίας για παραγωγή πολύπλοκων *emoji* συνθέσεων μέσω *reinforcement learning*, αλλά ταυτόχρονα αναδεικνύει την ανάγκη για περαιτέρω εκπαίδευση και ενίσχυση της συνάρτησης ανταμοιβής για επίτευξη πιο στιβαρής συμπεριφοράς υπό *deterministic* συνθήκες.

# Συμπεράσματα και Μελλοντική Εργασία

Η εργασία αυτή είχε ως στόχο τη δημιουργία ενός συστήματος που μετατρέπει φυσική γλώσσα σε σύνθετες εικόνες αποτελούμενες από *emojis*, συνδυάζοντας ασαφή λογική, γενετικούς αλγορίθμους και βαθιά ενισχυτική μάθηση. Το σύστημα εφαρμόστηκε επιτυχώς σε ένα συγκεκριμένο σενάριο εισόδου, στο οποίο χρησιμοποιείται ένα *emoji* ως πυρήνας, ένα *modifier* και ένα *accessory*, τοποθετημένα σε προκαθορισμένο φόντο.

Η πειραματική αξιολόγηση έδειξε ότι η απόδοση του πράκτορα PPO εξαρτάται σημαντικά από την αρχιτεκτονική του νευρωνικού του δικτύου. Αρχιτεκτονικές όπως `mlp_128x2` και `mlp_128x3` εμφάνισαν υψηλότερη συνέπεια στις ανταμοιβές και ποιοτικότερες συνθέσεις εικόνας, σε σύγκριση με μικρότερες όπως `mlp_64x2` ή `mlp_32x2`. Ο πράκτορας κατάφερε να τοποθετήσει βασικά σύμβολα στις κατάλληλες θέσεις και με σωστό μέγεθος, αντανακλώντας σε κάποιο βαθμό τη σημασιολογική δομή του *prompt*.

Ωστόσο, υπάρχουν σαφείς περιορισμοί. Πρώτον, δεν υλοποιήθηκαν μηχανισμοί χρωματισμού των *emojis*, παρότι είχαν προβλεφθεί σε επίπεδο δράσεων. Ο πράκτορας αγνοεί πλήρως την πληροφορία του χρώματος και η συνάρτηση ανταμοιβής δεν αξιολογεί την ορθότητα χρωματισμένων συμβόλων. Επιπλέον, οι χωρικές σχέσεις μεταξύ συμβόλων (π.χ. "κρατάει", "φοράει") δεν εφαρμόστηκαν δυναμικά, παρά μόνο αξιολογούνται έμμεσα στην ανταμοιβή μέσω της σχετικής τους θέσης.

Δεύτερον, το σύστημα έχει εκπαιδευτεί και αξιολογηθεί σε πολύ συγκεκριμένη μορφή προτάσεων, με σταθερό πλήθος και ρόλους *emojis*. Ένα *prompt* που δεν περιλαμβάνει *modifier*, ή που περιέχει δύο *accessories* ή καμία σαφή έννοια πυρήνα, ενδέχεται να απαιτεί εντελώς διαφορετική αρχιτεκτονική ή συνάρτηση ανταμοιβής για να αντιμετωπιστεί σωστά. Συνεπώς, η γενίκευση του πράκτορα σε εναλλακτικές εισόδους παραμένει περιορισμένη.

Για μελλοντική εργασία, θα μπορούσε να εξεταστεί η βελτίωση της *reward function* με πιο ακριβή χωρικά κριτήρια, η υλοποίηση ενεργειών χρωματισμού, καθώς και η εκπαίδευση σε ευρύτερη ποικιλία προτάσεων.

Συνοψίζοντας, το *GenEmoji* απέδειξε την πρακτική δυνατότητα παραγωγής εξατομικευμένων συνθέσεων *emoji* βάσει φυσικής γλώσσας, υπό την προϋπόθεση ότι η είσοδος είναι συμβατή με τη δομή για την οποία έχει εκπαιδευτεί ο πράκτορας. Οι δυνατότητες επέκτασης του συστήματος είναι πολλές και θα μπορούσαν να οδηγήσουν σε πιο δημιουργικές, εκφραστικές και εύχρηστες εφαρμογές.

## Οδηγίες Εκτέλεσης Πειραμάτων

Η παρούσα ενότητα περιγράφει τα απαιτούμενα βήματα για την αναπαραγωγή των αποτελεσμάτων του συστήματος *GenEmoji*, καθώς και την εκπαίδευση και αξιολόγηση των πρακτόρων PPO. Το σύστημα αναπτύχθηκε και δοκιμάστηκε σε περιβάλλον *Python 3.10* με χρήση του *PyTorch* και της βιβλιοθήκης *stable-baselines3*.

### Προαπαιτούμενα

#### Ελάχιστες απαιτήσεις:

- Python 3.10+
- 8GB RAM
- Επεξεργαστής με υποστήριξη AVX (για PyTorch σε CPU)



## Εγκατάσταση βιβλιοθηκών:

Εκτελέστε την εντολή:

```
pip install -r requirements.txt
```

Αν δεν υπάρχει κατεβάστε το μοντέλο *SpaCy* (μόνο μία φορά):

```
python -m spacy download en_core_web_sm
```

## Δομή Αρχείων

Φάκελος / Αρχείο	Περιγραφή
main.py	Εκπαίδευση όλων των πρακτόρων PPO
evaluate_and_plot.py	Αξιολόγηση πρακτόρων και αποθήκευση αποτελεσμάτων
tokenization.py	Ανάλυση prompt και fuzzy χαρακτηριστικών
emoji_generation.py	Γενετικός αλγόριθμος επιλογής emoji
image_generation.py	Περιβάλλον DRL και διαδικασία εκπαίδευσης/σύνθεσης
emoji_fuzzy_scores.json	Emoji dataset με fuzzy χαρακτηριστικά
emoji_images/	Εικόνες PNG ανά emoji, με βάση το Unicode
models/	Εκπαιδευμένα μοντέλα PPO (ανά αρχιτεκτονική)
results/	Αρχεία αξιολόγησης ανά πράκτορα (.csv, .png)

## Εκτέλεση Εκπαίδευσης

Επεξεργαστείτε ή επιλέξτε *prompt* μέσα στο *main.py*.

Εκτελέστε:

```
python main.py
```

Αυτό θα εκπαιδεύσει πράκτορες PPO για 6 διαφορετικές MLP αρχιτεκτονικές και θα αποθηκεύσει τα μοντέλα στο φάκελο *models/*.

## Εκτέλεση Αξιολόγησης

Αφού ολοκληρωθεί η εκπαίδευση, τρέξτε:

```
python evaluate_and_plot.py
```

Για κάθε πράκτορα:

- Τρέχουν 10 δοκιμαστικά επεισόδια
- Αποθηκεύονται:

- *.png* εικόνες με τη σύνθεση
- *rewards.csv* με ανταμοιβές
- *reward\_plot.png* με γράφημα ανταμοιβών

Τα αποτελέσματα αποθηκεύονται στον φάκελο:  
models/<όνομα\_αρχιτεκτονικής>/eval/

### Παρατηρήσεις

- Ο αλγόριθμος PPO μαθαίνει από το μηδέν χωρίς προεκπαίδευση.
- Οι πράκτορες μπορούν να τροποποιηθούν για νέα *prompts* αλλά είναι βελτιστοποιημένοι για συγκεκριμένο πλήθος και είδος *emojis*.
- Ο χρωματισμός και η δυναμική χωρική τοποθέτηση έχουν οριστεί σε επίπεδο περιβάλλοντος αλλά δεν ενεργοποιούνται πλήρως στο παρόν πείραμα.